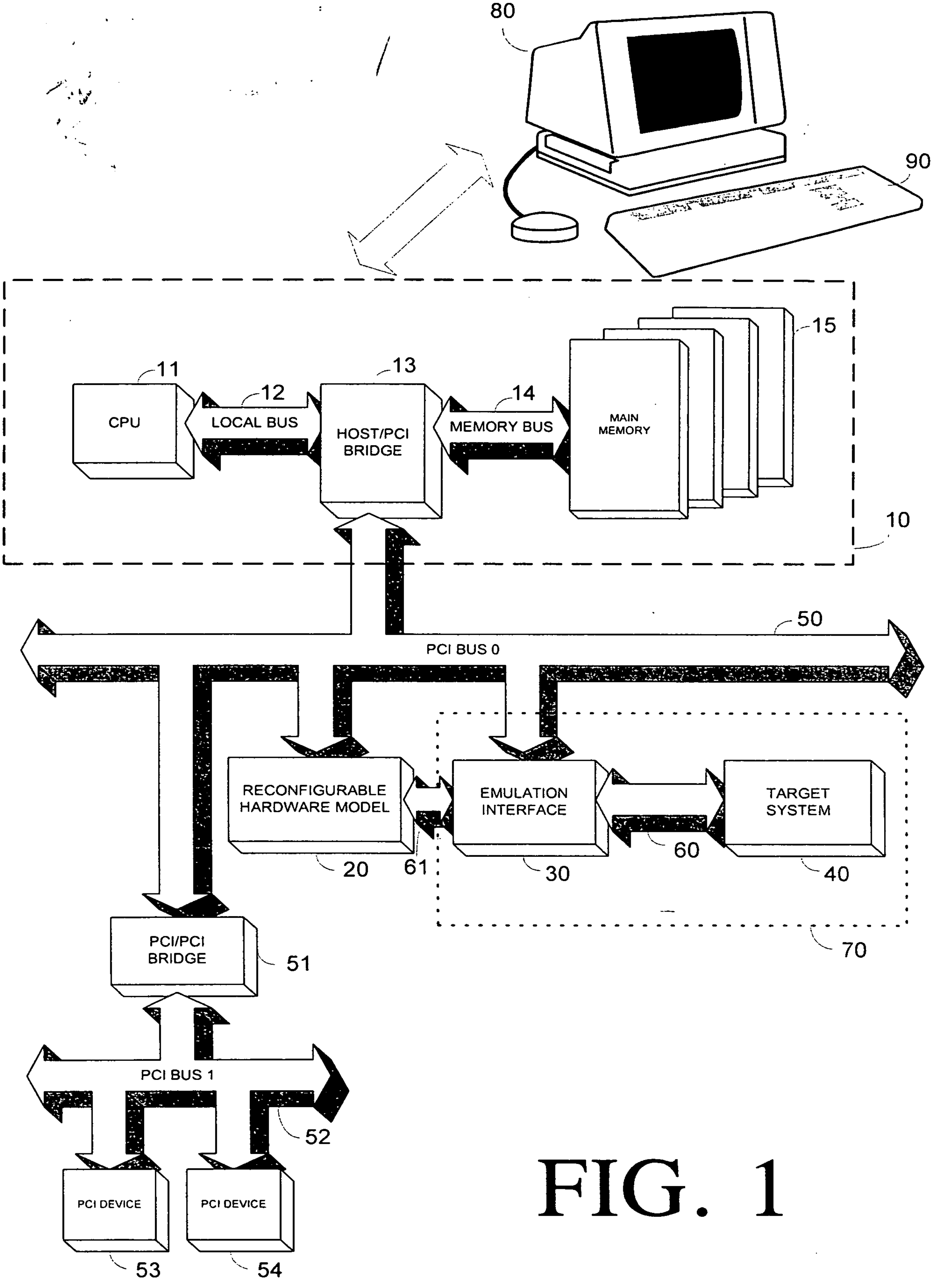
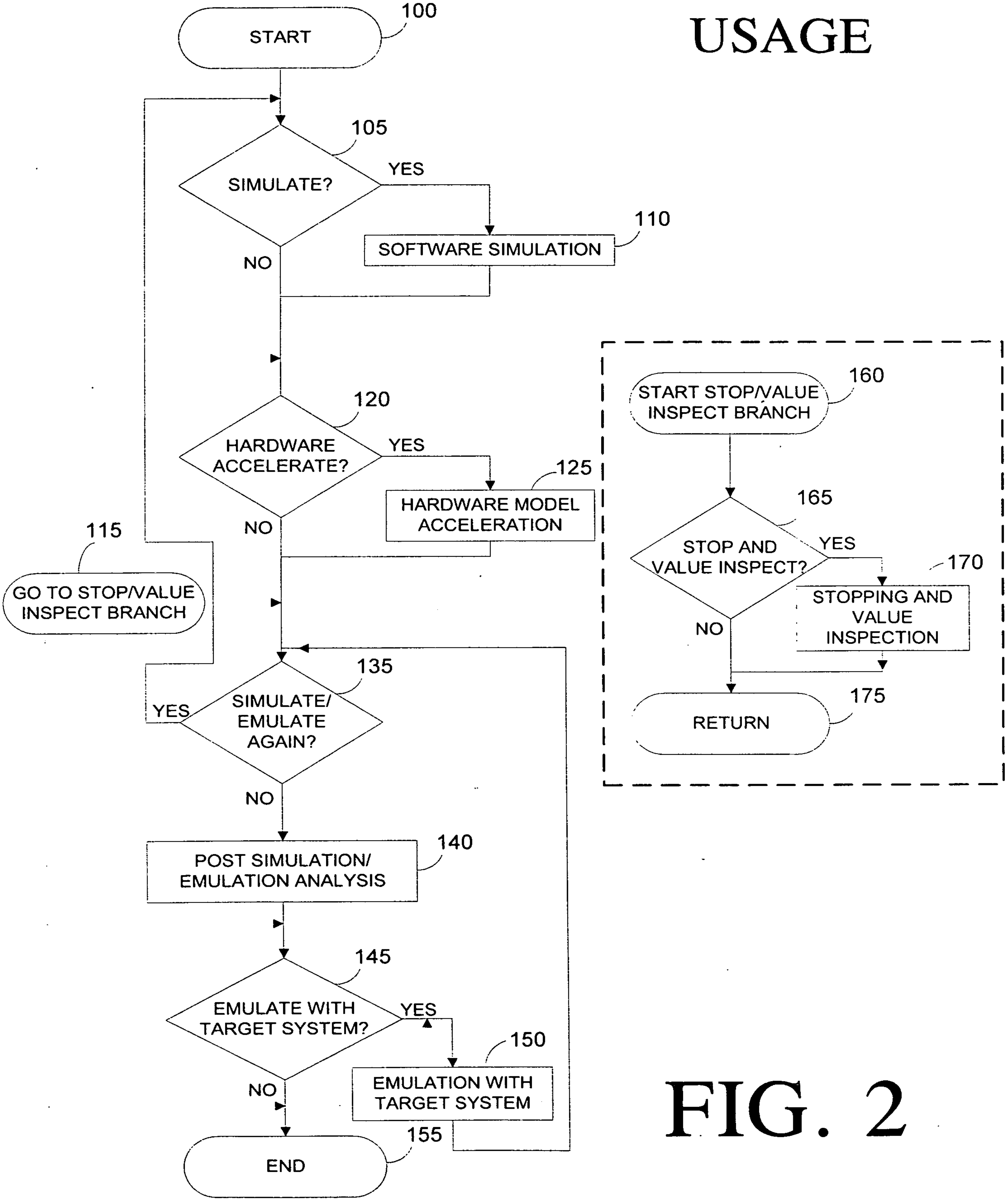

80
90
15
11
CPU
12
LOCAL BUS
13
HOST/PCI BRIDGE
14
MEMORY BUS
MAIN MEMORY
10
50
PCI BUS 0
RECONFIGURABLE HARDWARE MODEL
EMULATION INTERFACE
TARGET SYSTEM
20
61
30
60
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 1

USAGE



START
100
SIMULATE?
105
YES
NO
SOFTWARE SIMULATION
110
HARDWARE ACCELERATE?
120
YES
NO
HARDWARE MODEL ACCELERATION
125
115
GO TO STOP/VALUE INSPECT BRANCH
SIMULATE/ EMULATE AGAIN?
135
YES
NO
POST SIMULATION/ EMULATION ANALYSIS
140
EMULATE WITH TARGET SYSTEM?
145
YES
NO
EMULATION WITH TARGET SYSTEM
150
END
155
START STOP/VALUE INSPECT BRANCH
160
STOP AND VALUE INSPECT?
165
YES
NO
STOPPING AND VALUE INSPECTION
170
RETURN
175

FIG. 2

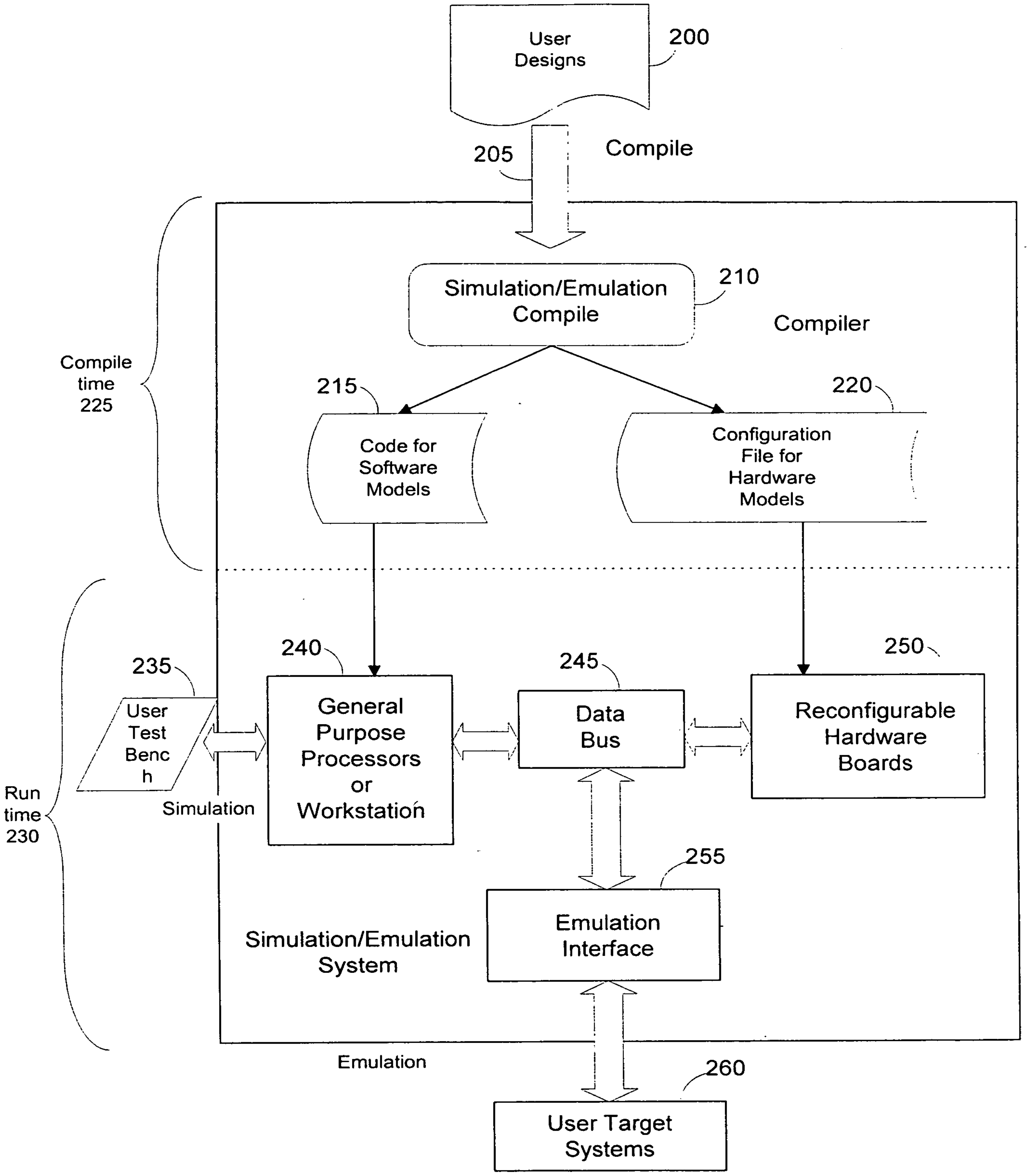

User Designs
200
205
Compile
Simulation/Emulation Compile
210
Compiler
Compile time 225
215
Code for Software Models
220
Configuration File for Hardware Models
235
User Test Benc h
Simulation
240
General Purpose Processors or Workstation
245
Data Bus
250
Reconfigurable Hardware Boards
Run time 230
255
Emulation Interface
Simulation/Emulation System
Emulation
260
User Target Systems

FIG. 3

COMPILATION

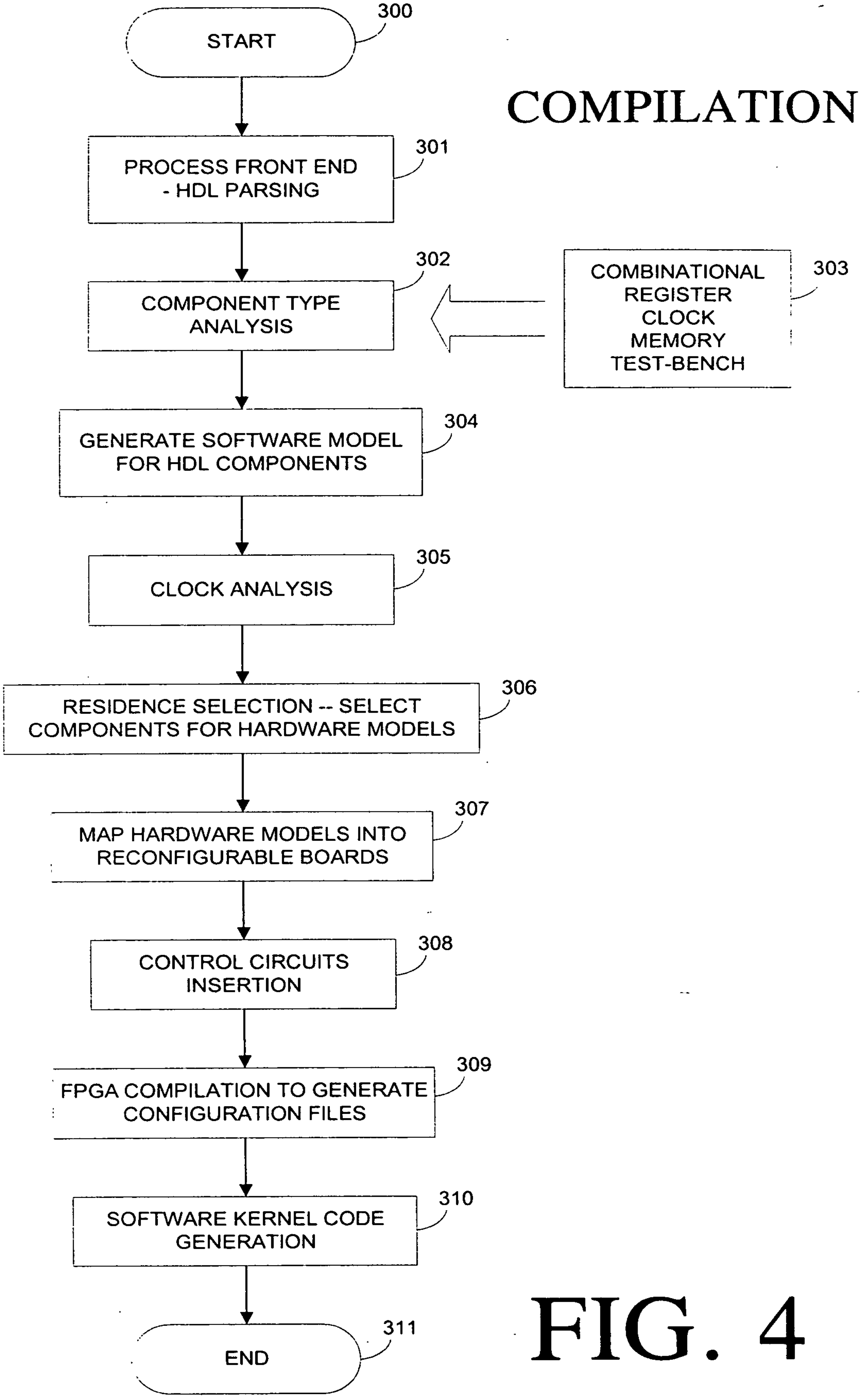

START
300
PROCESS FRONT END - HDL PARSING
301
COMPONENT TYPE ANALYSIS
302
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
303
GENERATE SOFTWARE MODEL FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION -- SELECT COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS INSERTION
308
FPGA COMPILATION TO GENERATE CONFIGURATION FILES
309
SOFTWARE KERNEL CODE GENERATION
310
END
311

FIG. 4

SOFTWARE KERNEL

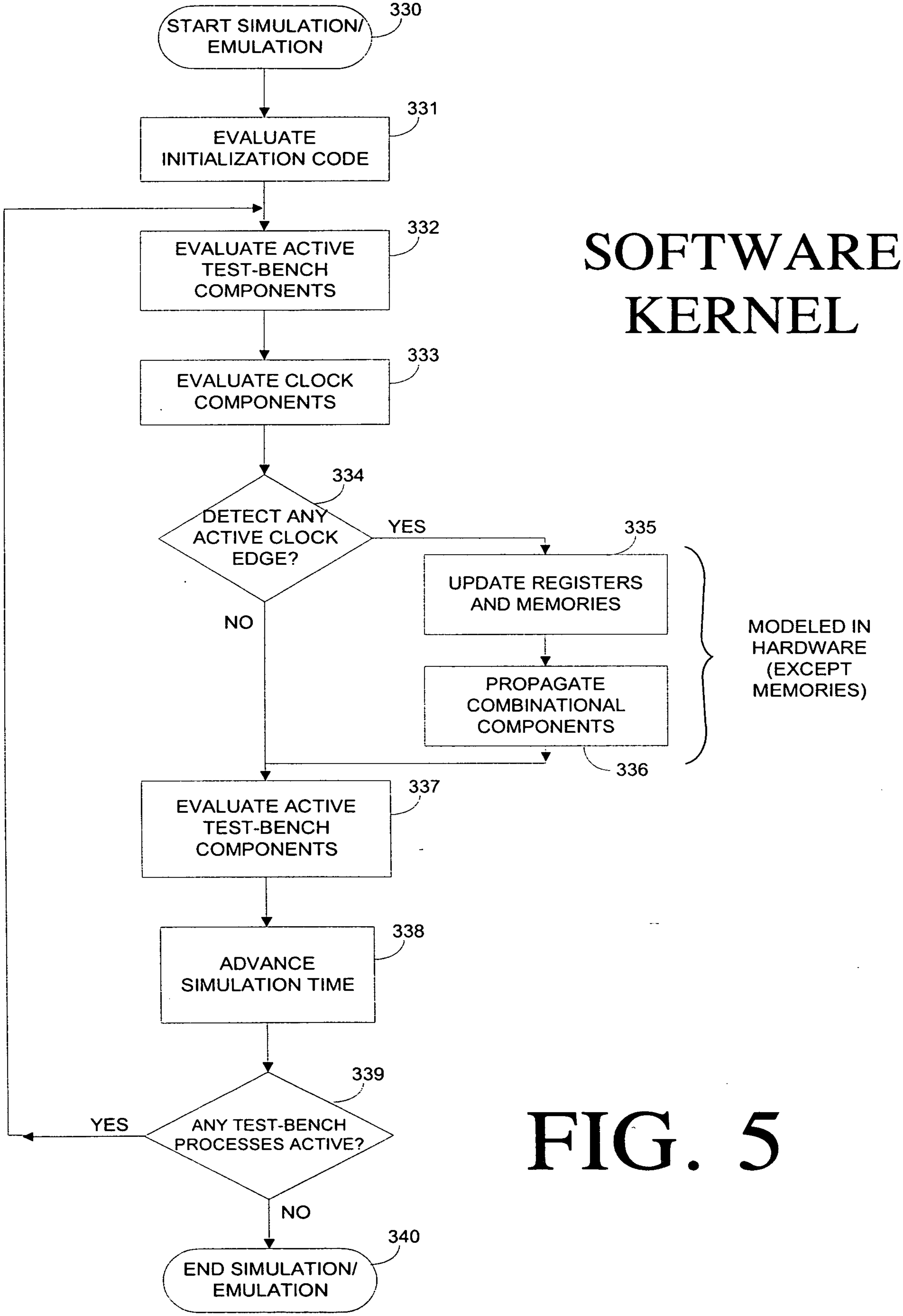

START SIMULATION/ EMULATION
330
EVALUATE INITIALIZATION CODE
331
EVALUATE ACTIVE TEST-BENCH COMPONENTS
332
EVALUATE CLOCK COMPONENTS
333
DETECT ANY ACTIVE CLOCK EDGE?
334
YES
NO
UPDATE REGISTERS AND MEMORIES
335
PROPAGATE COMBINATIONAL COMPONENTS
336
MODELED IN HARDWARE (EXCEPT MEMORIES)
EVALUATE ACTIVE TEST-BENCH COMPONENTS
337
ADVANCE SIMULATION TIME
338
ANY TEST-BENCH PROCESSES ACTIVE?
339
YES
NO
END SIMULATION/ EMULATION
340

FIG. 5

MAPPING HARDWARE MODELS TO RECONFIGURABLE BOARDS

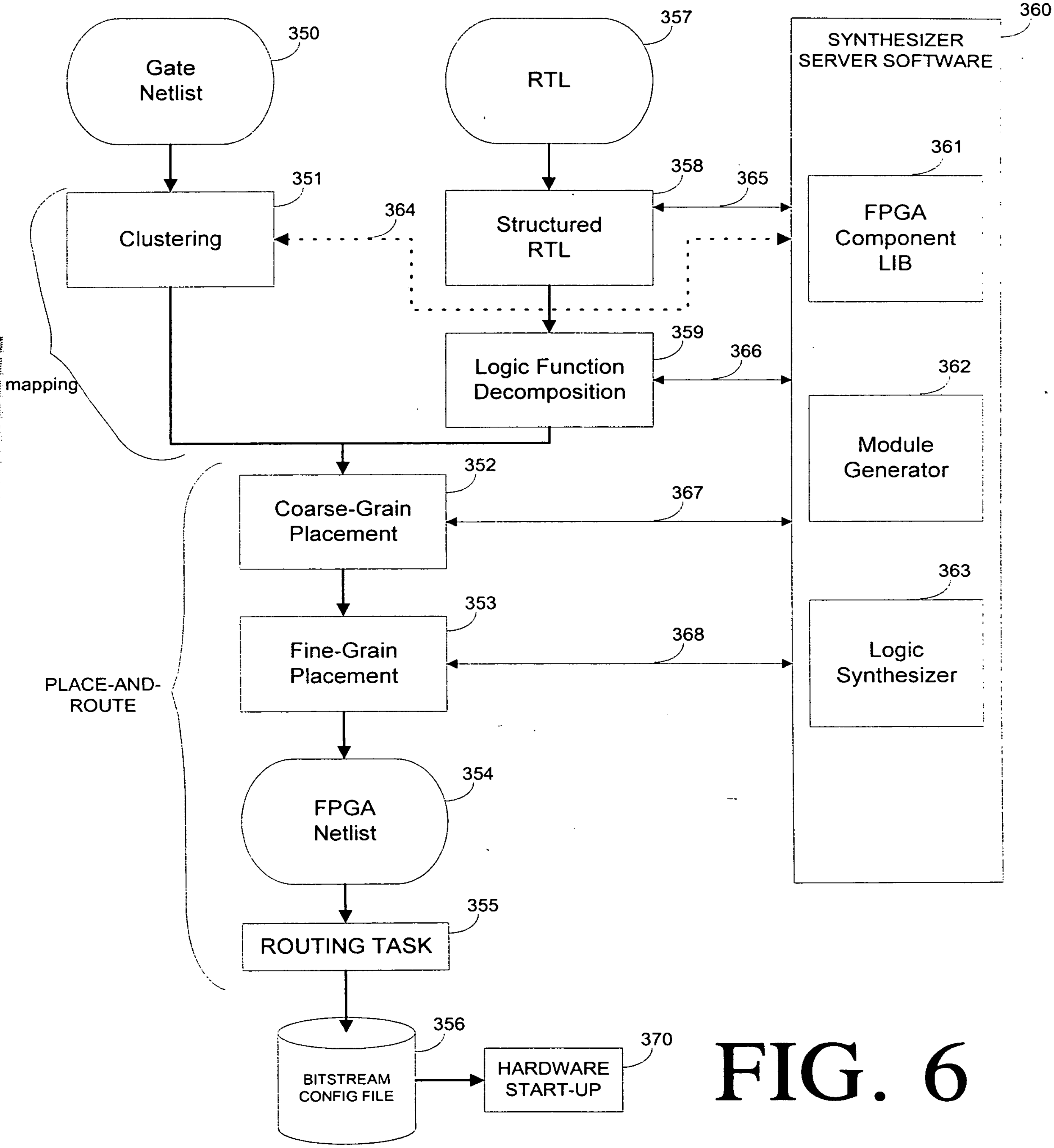
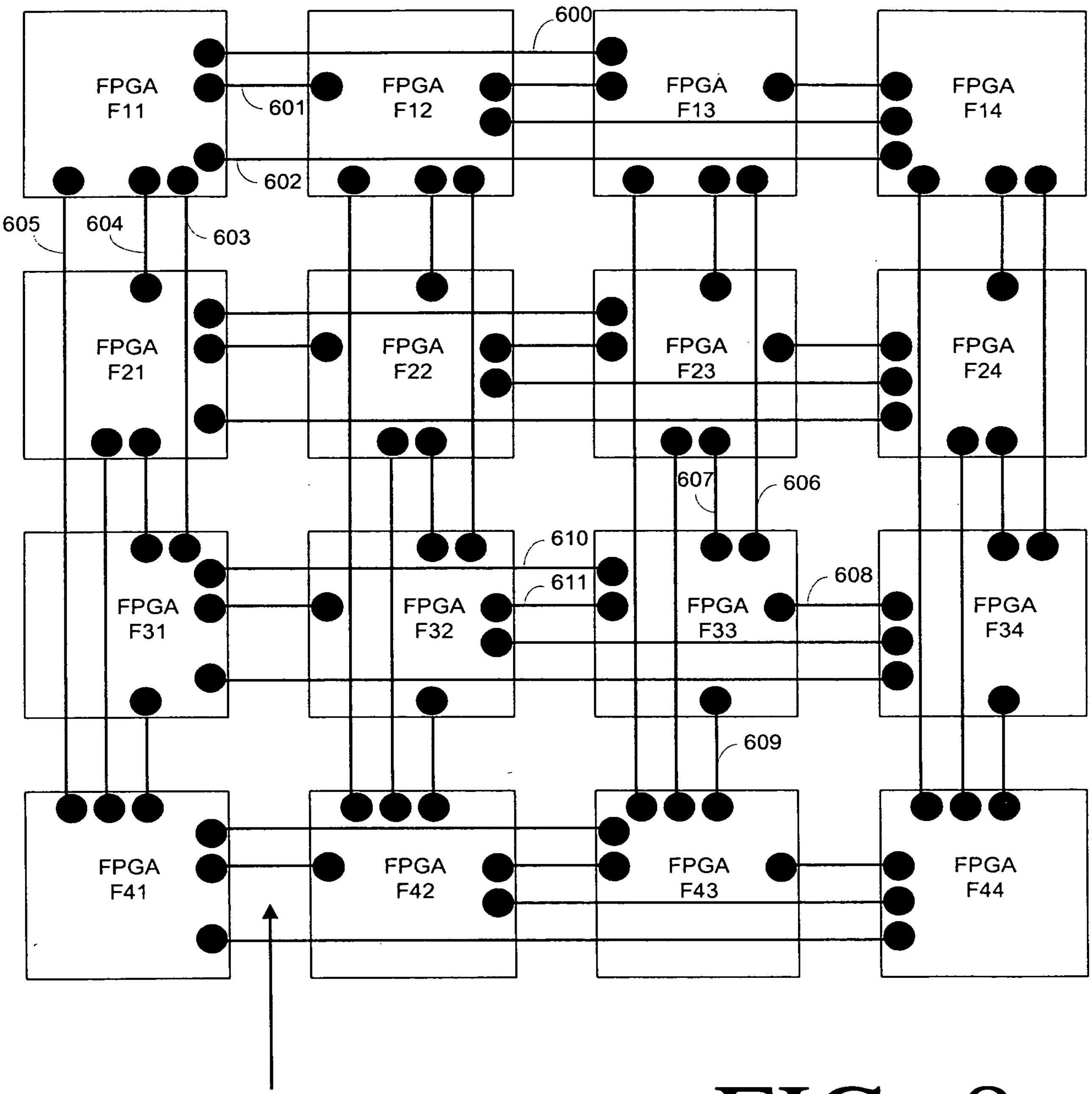


FIG. 6

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	1	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	1	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	1	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

FPGA INTERCONNECTION



1/6 of total I/O pins of FPGA for interconnection

FIG. 8

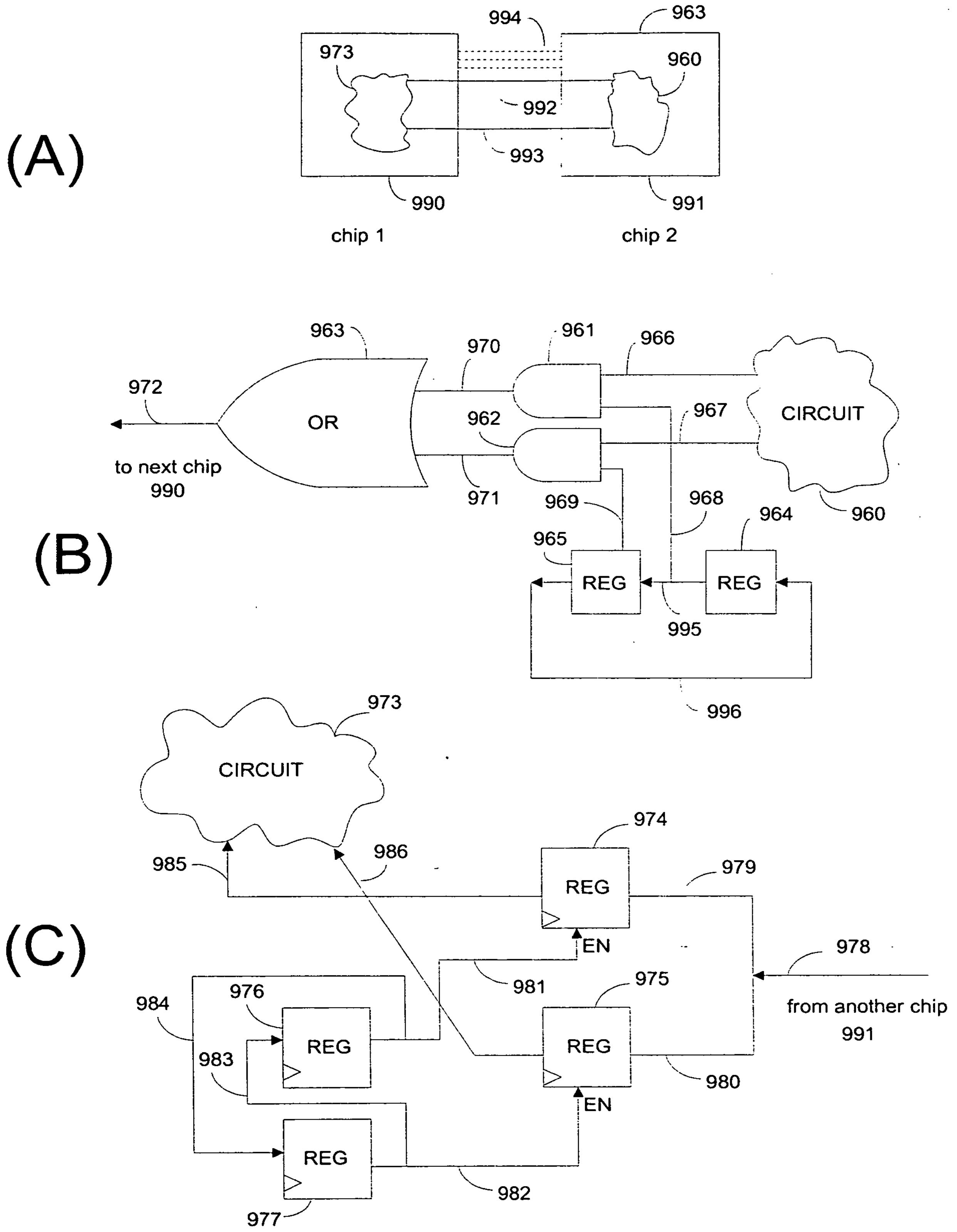

(A)
994
963
973
960
992
993
990
991
chip 1
chip 2
(B)
963
970
961
966
972
OR
962
967
CIRCUIT
to next chip
990
971
969
968
964
960
965
REG
REG
995
996
(C)
973
CIRCUIT
974
985
986
REG
979
EN
978
981
975
984
976
from another chip
991
983
REG
REG
980
EN
REG
982
977

FIG. 9

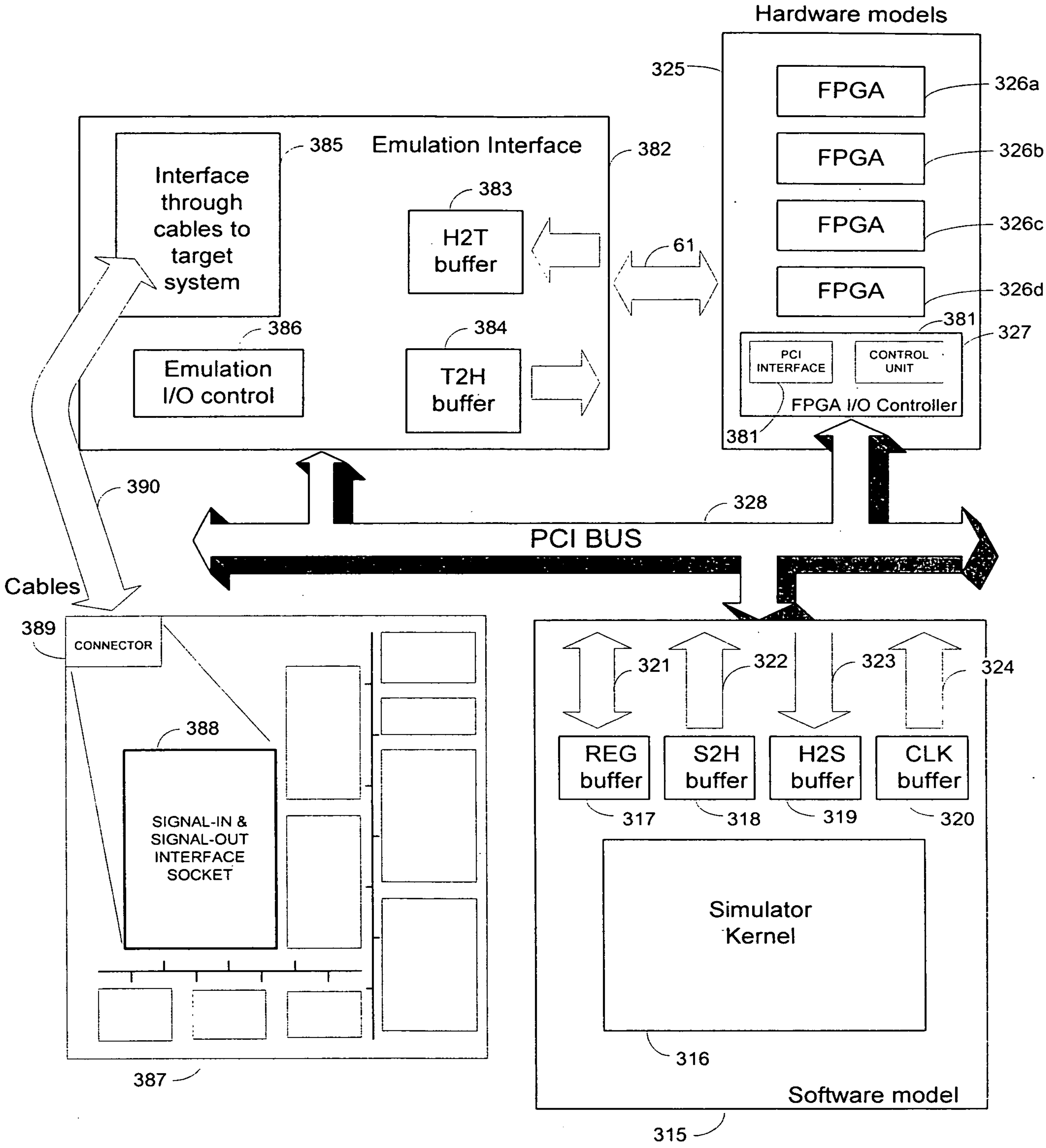

Hardware models
325
FPGA
326a
FPGA
326b
FPGA
326c
FPGA
326d
381
327
PCI INTERFACE
CONTROL UNIT
FPGA I/O Controller
381
385
Emulation Interface
382
Interface through cables to target system
383
H2T buffer
61
386
Emulation I/O control
384
T2H buffer
390
328
PCI BUS
Cables
389
CONNECTOR
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
387
321
322
323
324
REG buffer
S2H buffer
H2S buffer
CLK buffer
317
318
319
320
Simulator Kernel
316
Software model
315

FIG. 10

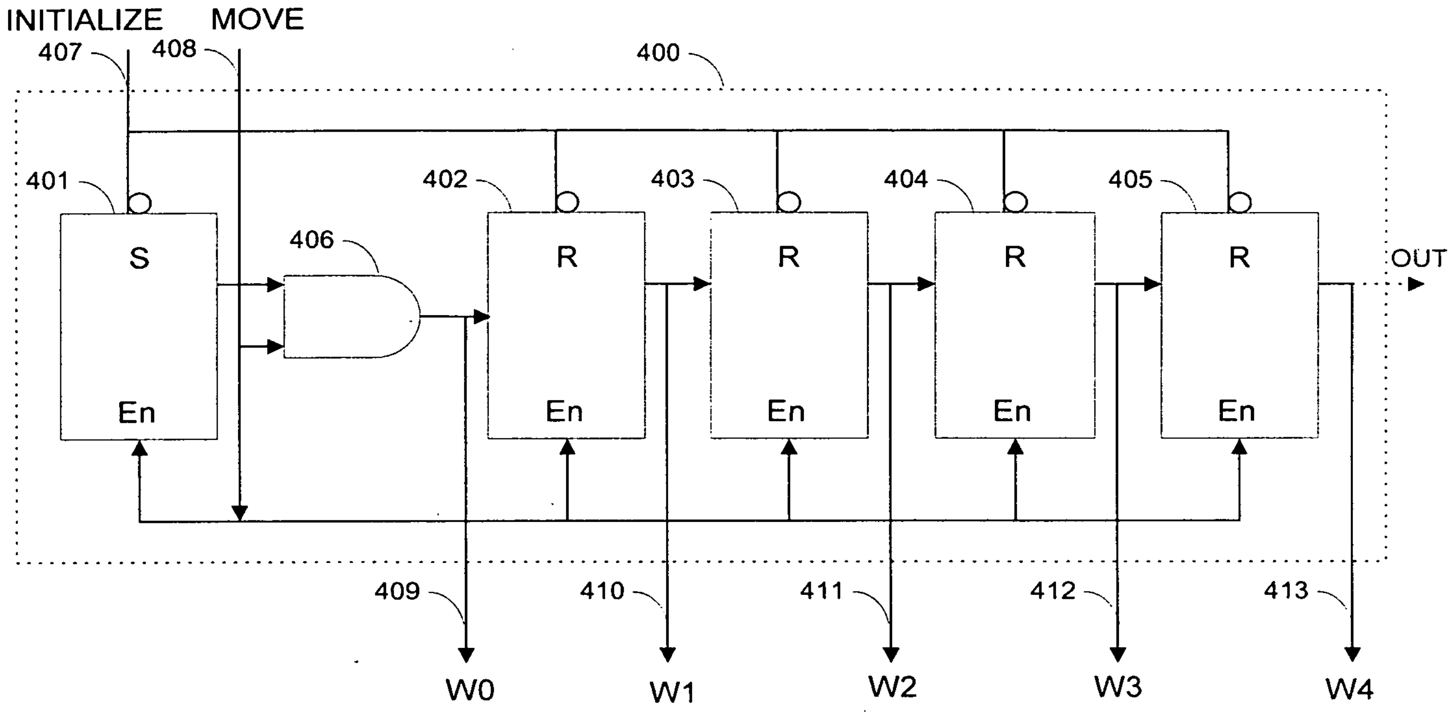

INITIALIZE
MOVE
407
408
400
401
402
403
404
405
406
S
R
R
R
R
En
En
En
En
En
OUT
409
410
411
412
413
W0
W1
W2
W3
W4

FIG. 11

ADDRESS POINTER INITIALIZATION

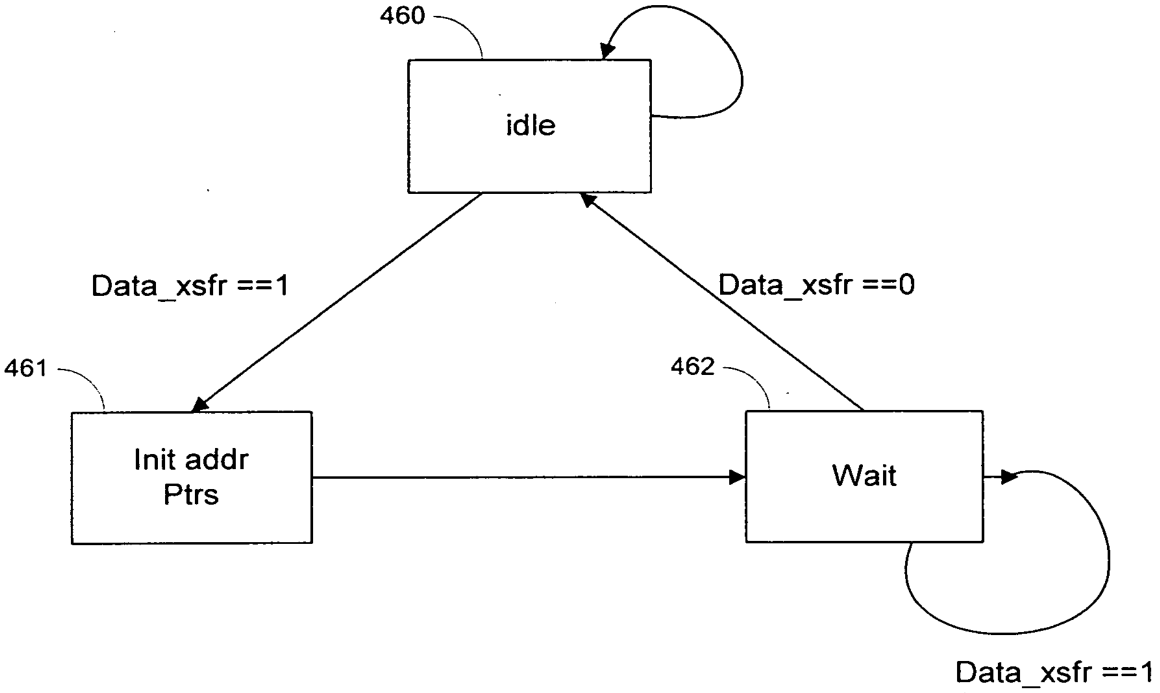


FIG. 12

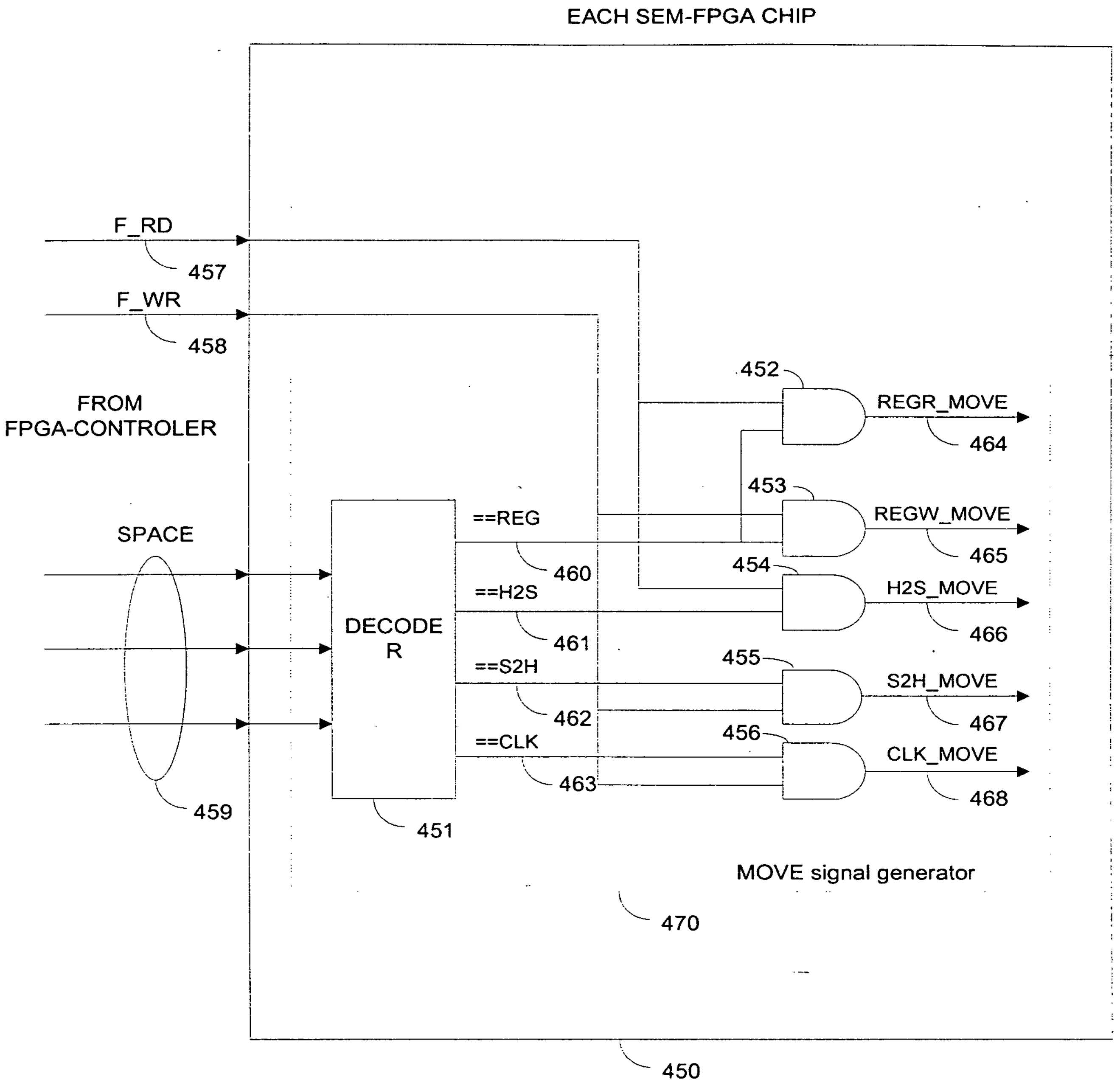

EACH SEM-FPGA CHIP
F_RD
457
F_WR
458
FROM
FPGA-CONTROLER
SPACE
459
DECODE
R
451
==REG
460
==H2S
461
==S2H
462
==CLK
463
452
453
454
455
456
REGR_MOVE
464
REGW_MOVE
465
H2S_MOVE
466
S2H_MOVE
467
CLK_MOVE
468
MOVE signal generator
470
450

FIG. 13

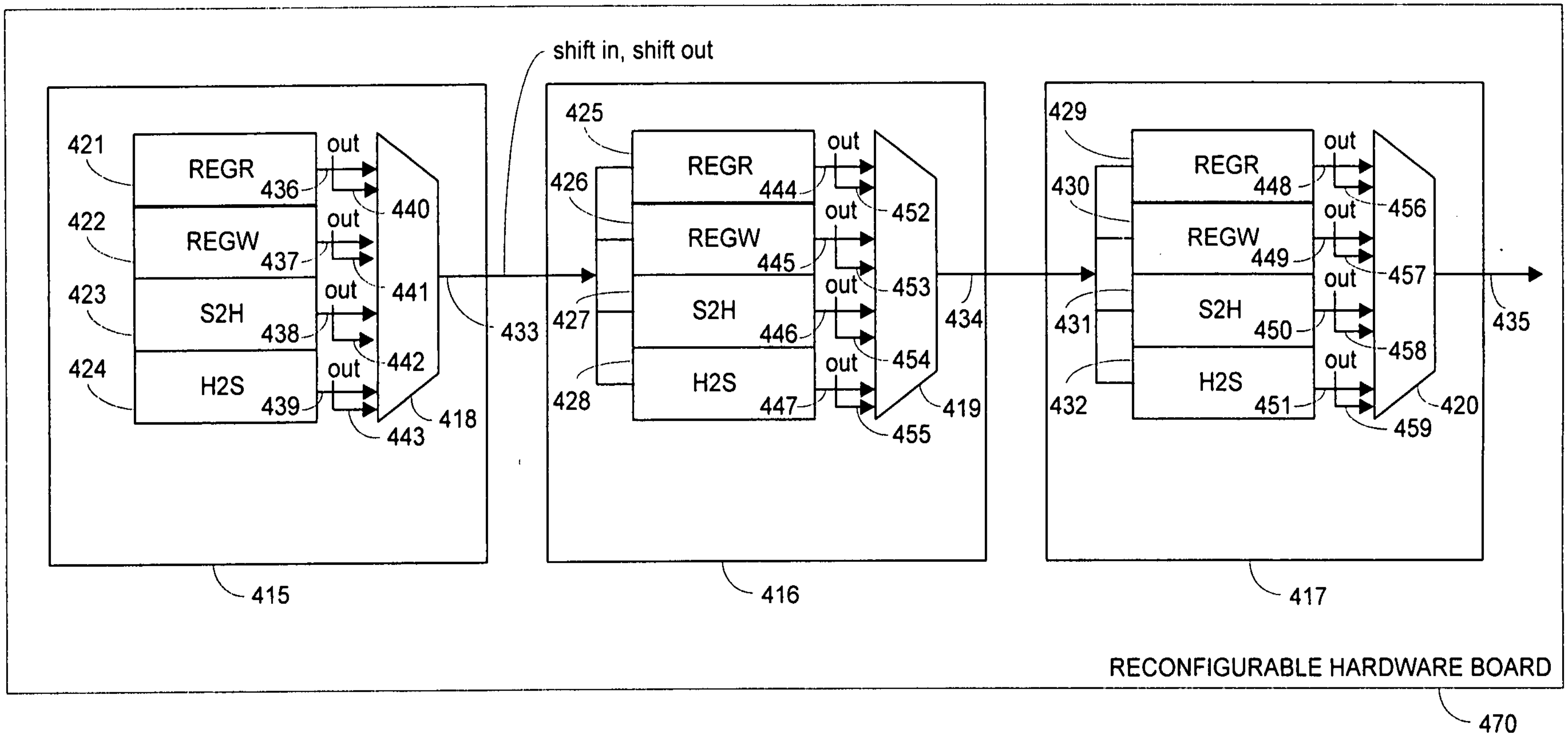

shift in, shift out
421
422
423
424
REGR
REGW
S2H
H2S
436
437
438
439
out
out
out
out
440
441
442
443
418
433
415
425
426
427
428
REGR
REGW
S2H
H2S
444
445
446
447
out
out
out
out
452
453
454
455
419
434
416
429
430
431
432
REGR
REGW
S2H
H2S
448
449
450
451
out
out
out
out
456
457
458
459
420
435
417
RECONFIGURABLE HARDWARE BOARD
470

FIG. 14

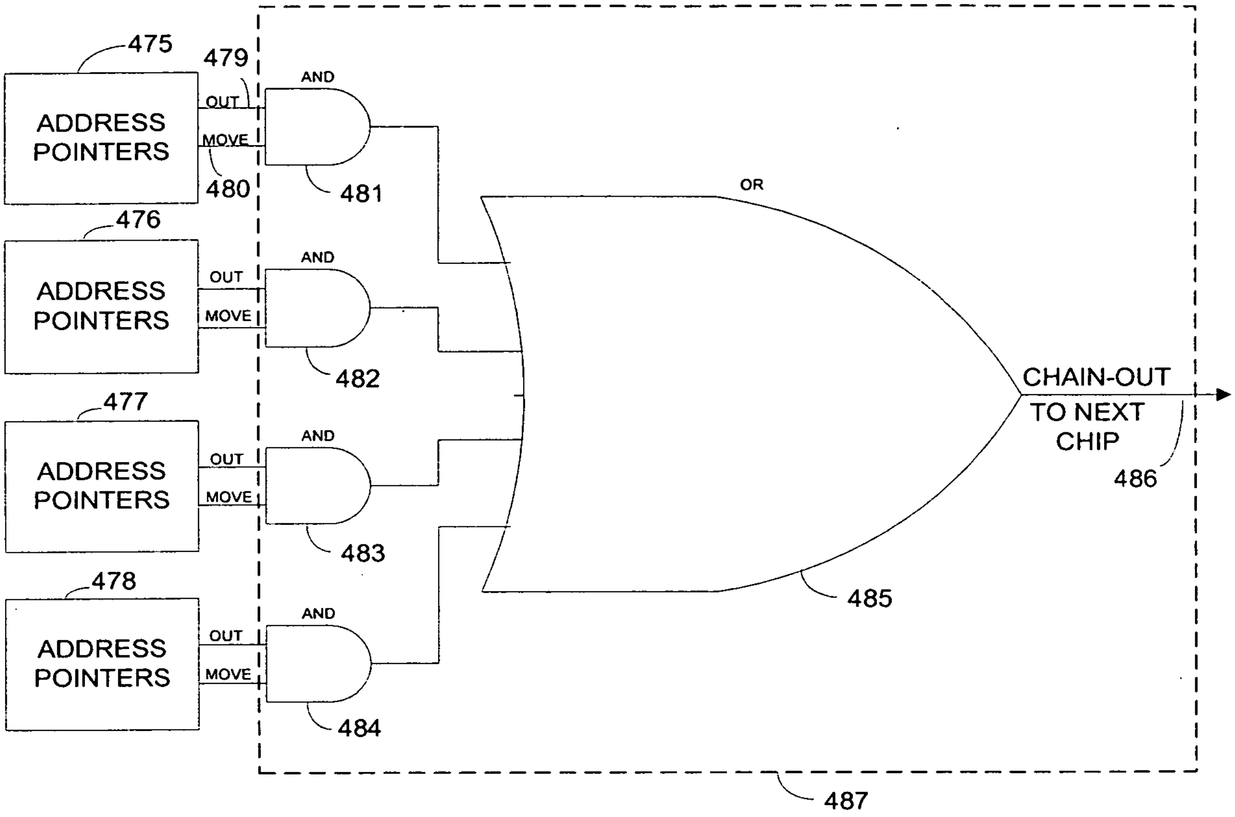
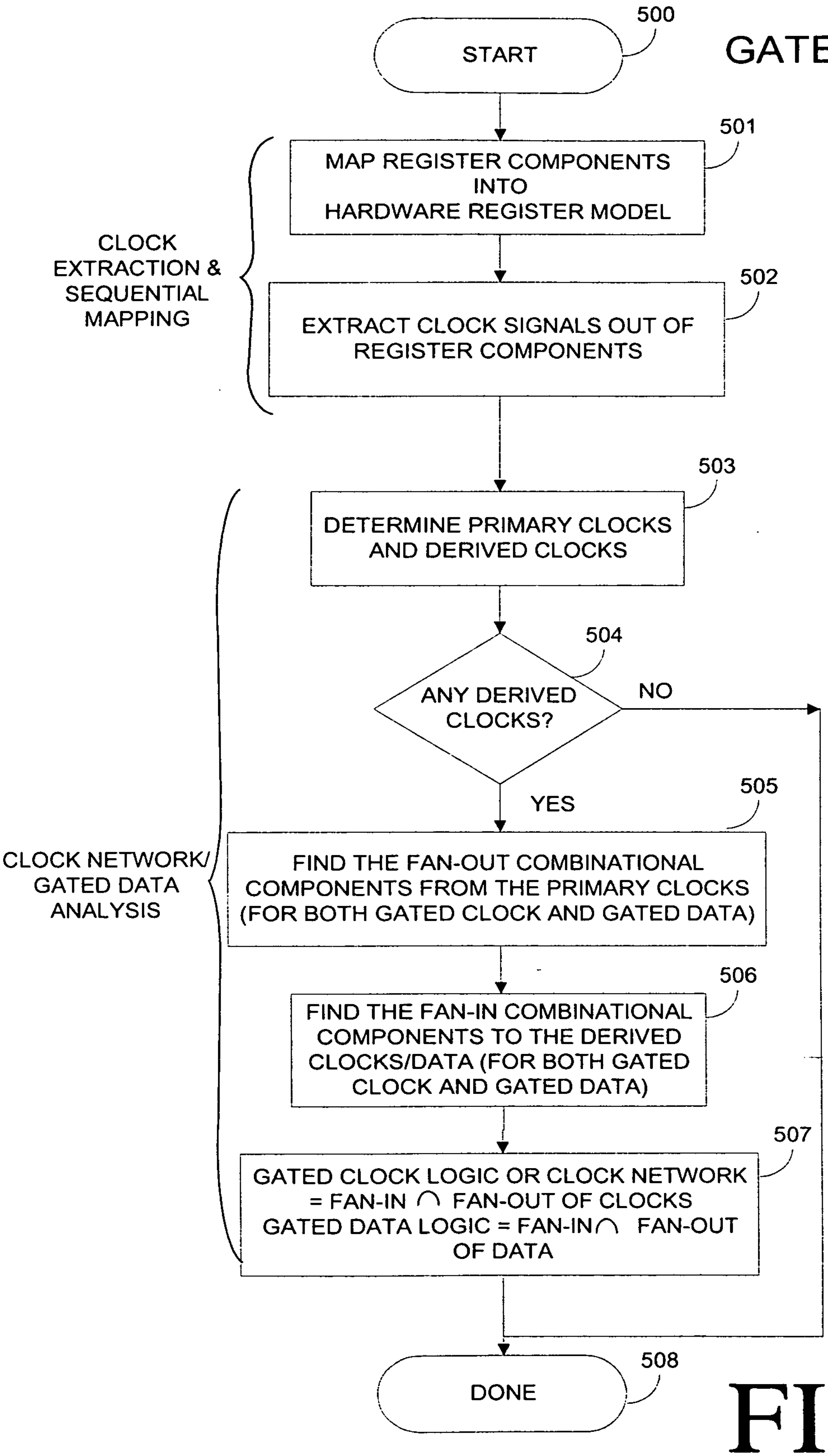

475
479
ADDRESS POINTERS
OUT
MOVE
480
AND
481
OR
476
ADDRESS POINTERS
OUT
MOVE
AND
482
477
ADDRESS POINTERS
OUT
MOVE
AND
483
CHAIN-OUT
TO NEXT CHIP
486
485
478
ADDRESS POINTERS
OUT
MOVE
AND
484
487

FIG. 15

GATED DATA/CLOCK ANALYSIS



500
START
501
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
CLOCK EXTRACTION & SEQUENTIAL MAPPING
502
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
503
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
504
ANY DERIVED CLOCKS?
NO
YES
505
CLOCK NETWORK/ GATED DATA ANALYSIS
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
506
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
507
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN ∩ FAN-OUT OF DATA
508
DONE

FIG. 16

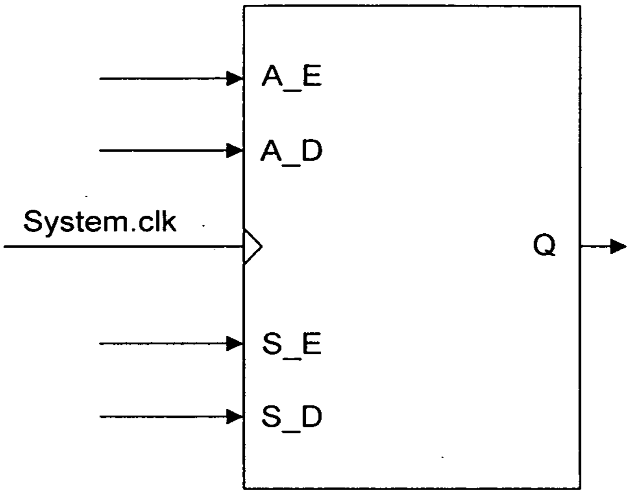
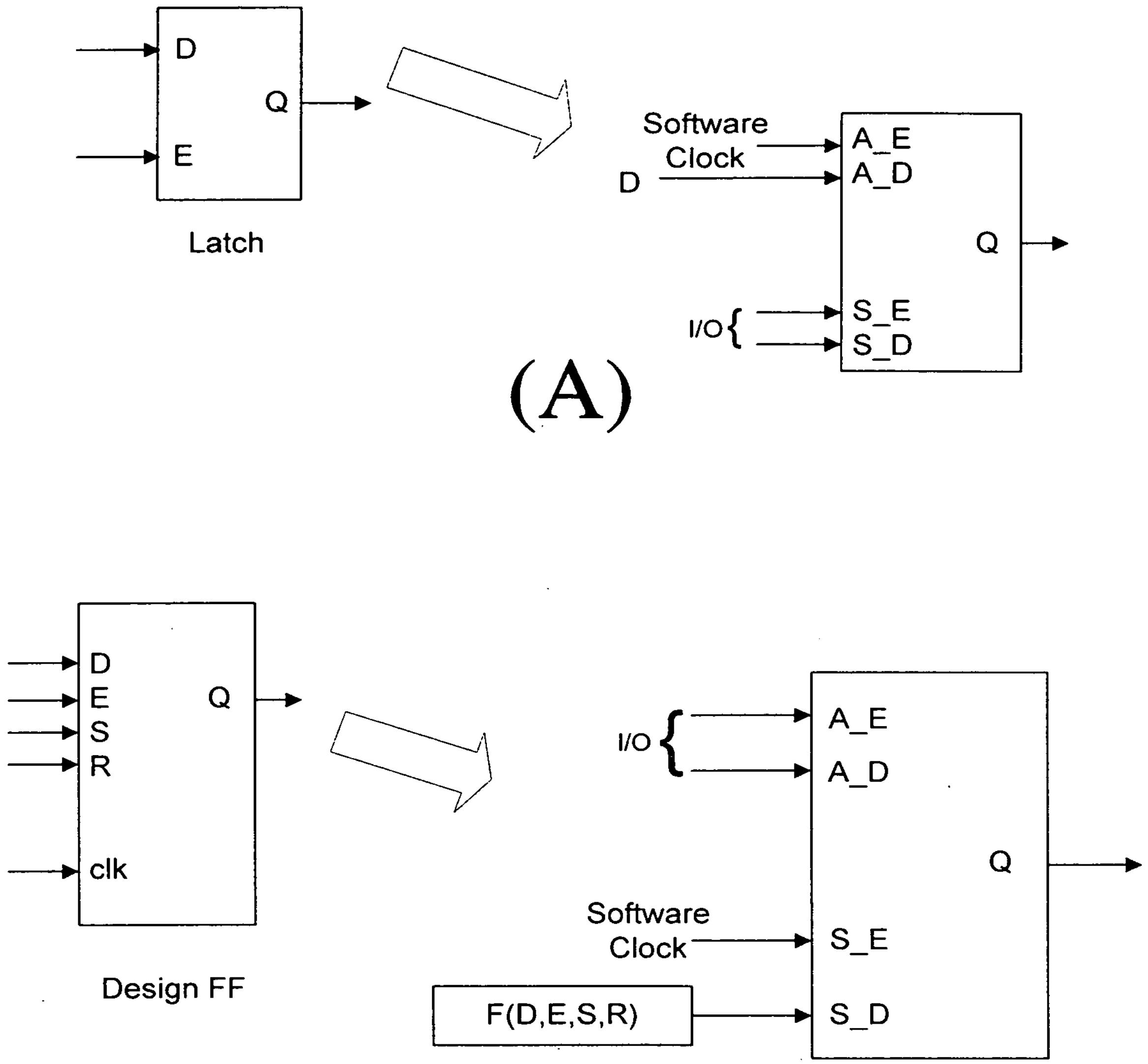


FIG. 17


D
E
Q
Latch
Software
Clock
D
A_E
A_D
Q
I/O
S_E
S_D
D
E
S
R
clk
Q
Design FF
I/O
A_E
A_D
Q
Software
Clock
S_E
F(D,E,S,R)
S_D

(A)

(B)

FIG. 18

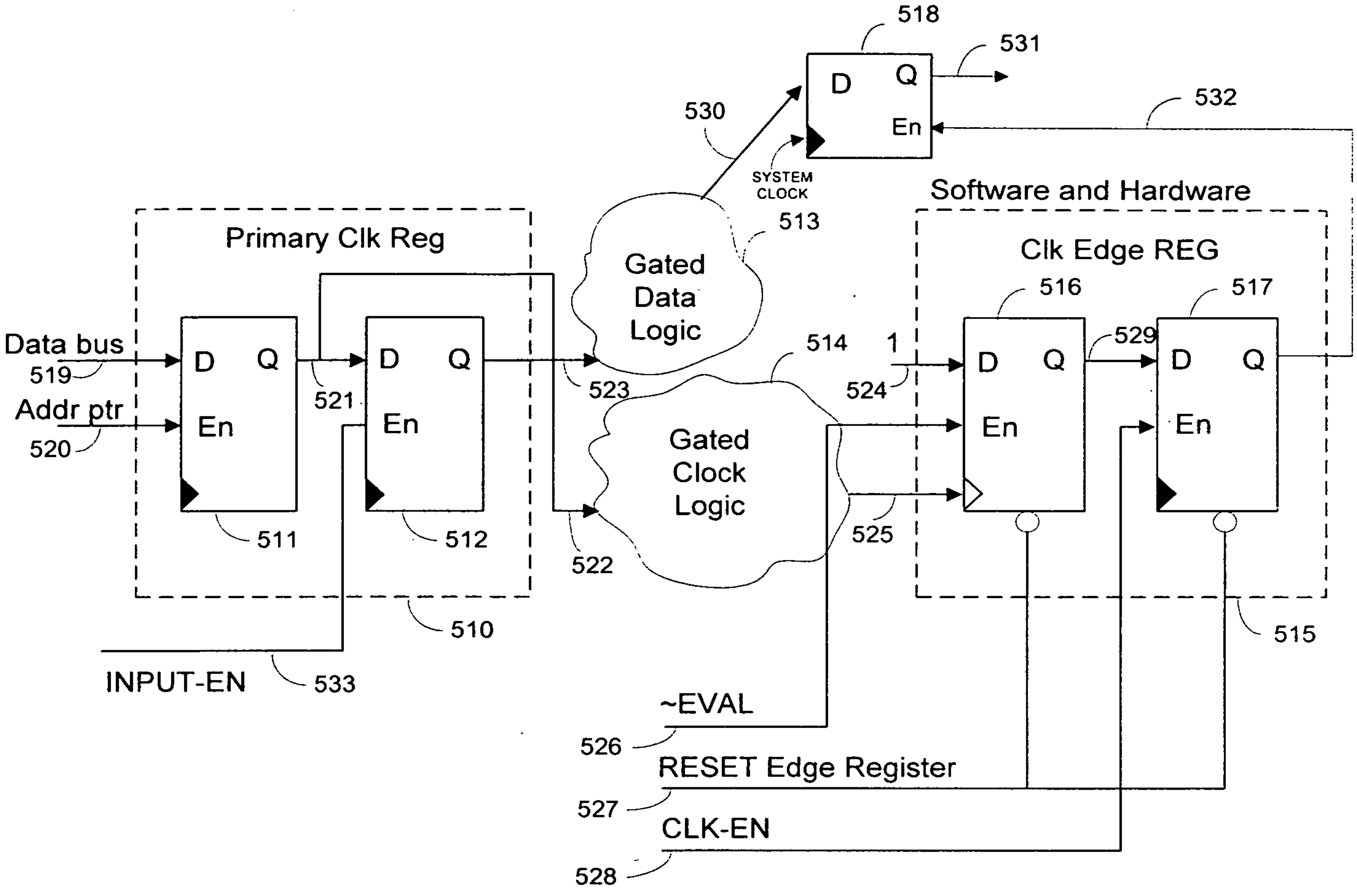

518
531
532
530
D
Q
En
SYSTEM CLOCK
513
Software and Hardware
Primary Clk Reg
Gated Data Logic
Clk Edge REG
516
517
Data bus
519
Addr ptr
520
D
Q
En
521
D
Q
En
523
514
1
524
529
D
Q
En
D
Q
En
Gated Clock Logic
511
512
522
525
510
515
INPUT-EN
533
~EVAL
526
RESET Edge Register
527
CLK-EN
528

FIG. 19

DURING
EVALUATION

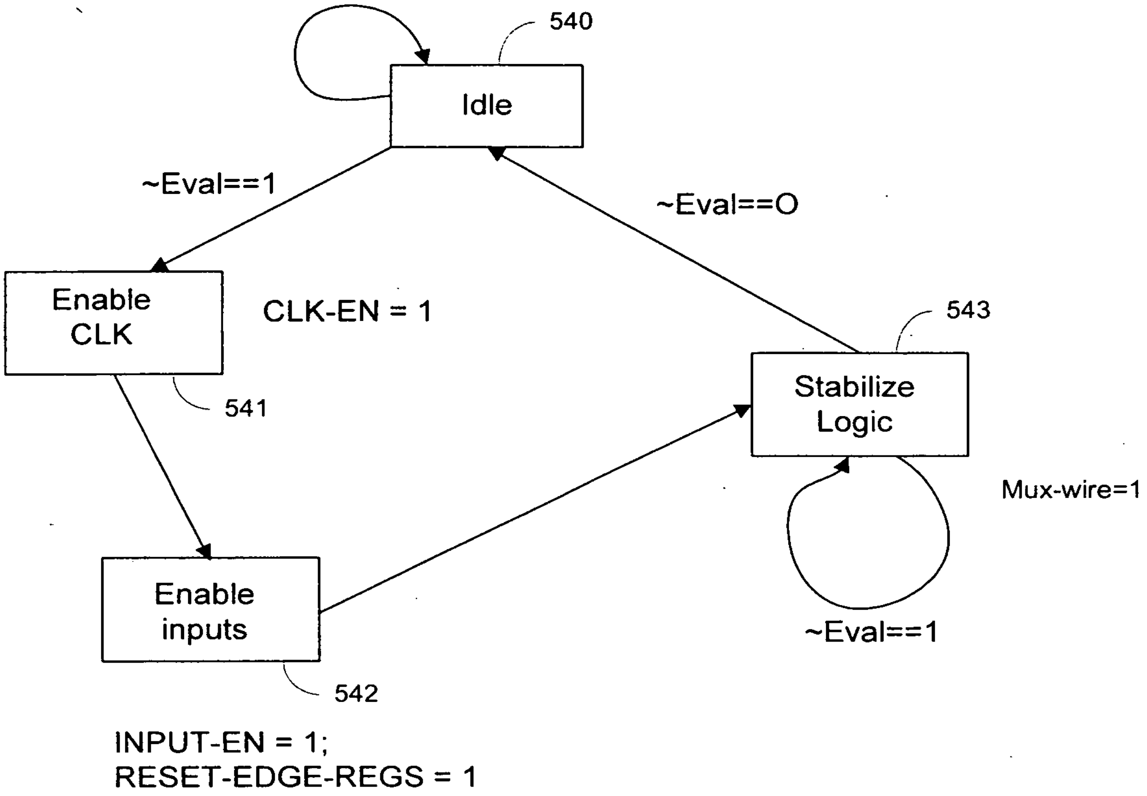


FIG. 20

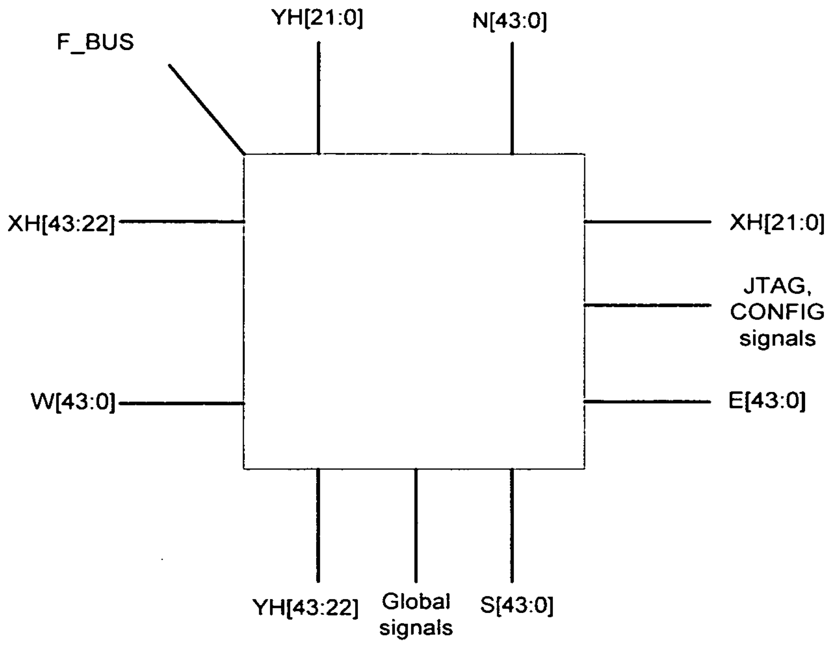
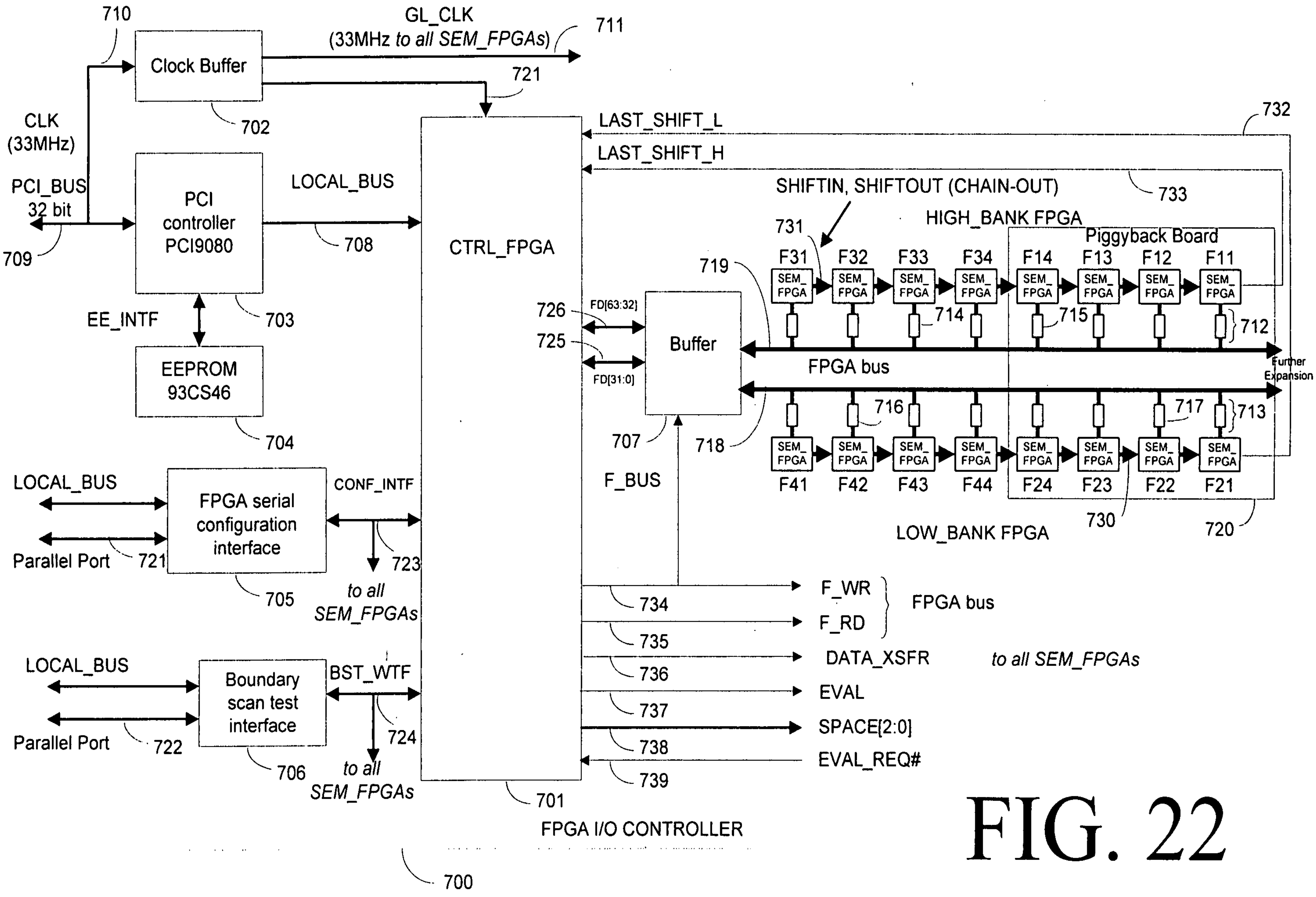

F_BUS
YH[21:0]
N[43:0]
XH[43:22]
XH[21:0]
JTAG, CONFIG signals
W[43:0]
E[43:0]
YH[43:22]
Global signals
S[43:0]

FIG. 21



710
Clock Buffer
702
GL_CLK
(33MHz to all SEM_FPGAs)
711
721
CLK
(33MHz)
PCI_BUS
32 bit
709
PCI
controller
PCI9080
703
LOCAL_BUS
708
EE_INTF
EEPROM
93CS46
704
CTRL_FPGA
LAST_SHIFT_L
LAST_SHIFT_H
732
733
SHIFTIN, SHIFTOUT (CHAIN-OUT)
HIGH_BANK FPGA
Piggyback Board
731
F31
F32
F33
F34
F14
F13
F12
F11
SEM_FPGA
714
715
712
719
726
725
FD[63:32]
FD[31:0]
Buffer
FPGA bus
Further Expansion
716
717
713
707
718
F41
F42
F43
F44
F24
F23
F22
F21
730
720
LOW_BANK FPGA
F_BUS
LOCAL_BUS
Parallel Port
721
FPGA serial configuration interface
705
CONF_INTF
723
to all SEM_FPGAs
LOCAL_BUS
Parallel Port
722
Boundary scan test interface
706
BST_WTF
724
to all SEM_FPGAs
734
F_WR
735
F_RD
FPGA bus
736
DATA_XSFR
to all SEM_FPGAs
737
EVAL
738
SPACE[2:0]
739
EVAL_REQ#
701
FPGA I/O CONTROLLER
700

FIG. 22

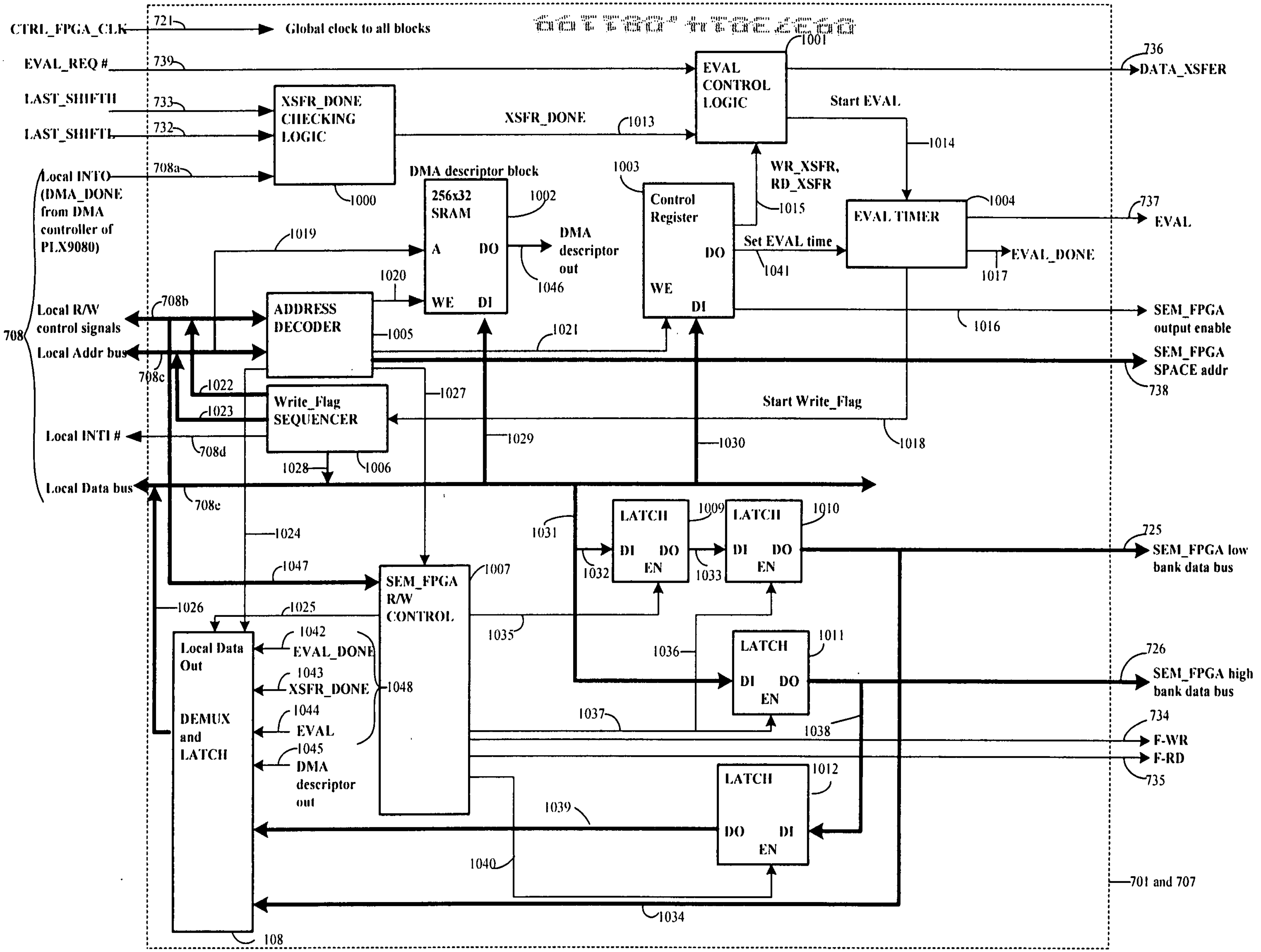

CTRL_FPGA_CLK
721
Global clock to all blocks
EVAL_REQ #
739
LAST_SHIFTH
733
LAST_SHIFTL
732
Local INTO (DMA_DONE from DMA controller of PLX9080)
708a
708
Local R/W control signals
708b
Local Addr bus
708c
Local INTI #
708d
Local Data bus
708e
XSFR_DONE CHECKING LOGIC
1000
XSFR_DONE
1013
EVAL CONTROL LOGIC
1001
DATA_XSFER
736
Start EVAL
1014
DMA descriptor block
256x32 SRAM
A
DO
WE
DI
1002
DMA descriptor out
1046
1019
1020
ADDRESS DECODER
1005
1021
1027
Write_Flag SEQUENCER
1006
1022
1023
1028
1029
1003
Control Register
DO
WE
DI
WR_XSFR, RD_XSFR
1015
Set EVAL time
1041
EVAL TIMER
1004
EVAL
737
EVAL_DONE
1017
1016
SEM_FPGA output enable
SEM_FPGA SPACE addr
738
Start Write_Flag
1018
1030
1024
1047
SEM_FPGA R/W CONTROL
1007
1025
1026
Local Data Out
DEMUX and LATCH
108
1042
EVAL_DONE
1043
XSFR_DONE
1044
EVAL
1045
DMA descriptor out
1048
1031
1032
LATCH
DI
DO
EN
1009
1033
LATCH
DI
DO
EN
1010
SEM_FPGA low bank data bus
725
1035
1036
LATCH
DI
DO
EN
1011
SEM_FPGA high bank data bus
726
1037
1038
F-WR
734
F-RD
735
LATCH
DO
DI
EN
1012
1039
1040
1034
701 and 707

FIG. 23

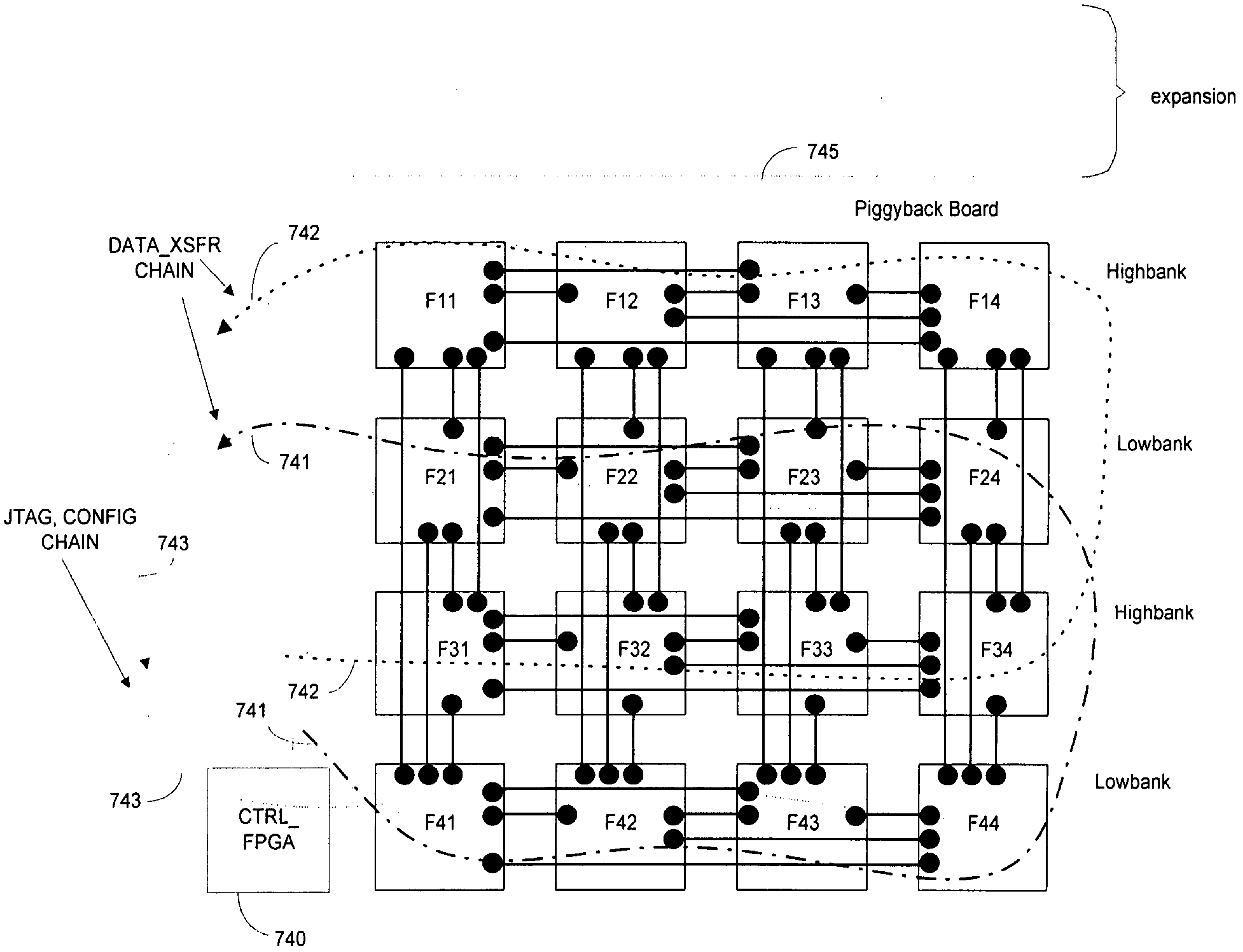

expansion
745
Piggyback Board
DATA_XSFR
CHAIN
742
F11
F12
F13
F14
Highbank
741
F21
F22
F23
F24
Lowbank
JTAG, CONFIG
CHAIN
743
F31
F32
F33
F34
Highbank
742
741
743
CTRL_
FPGA
F41
F42
F43
F44
Lowbank
740

FIG. 24

HARDWARE START-UP

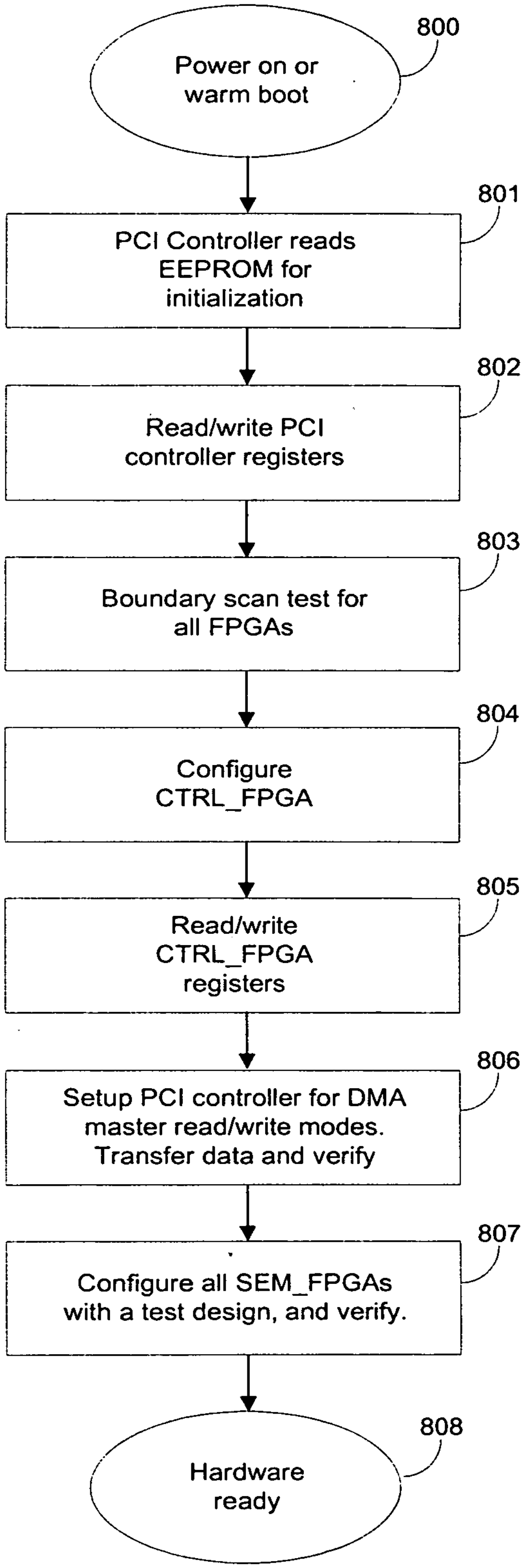

Power on or warm boot
800
PCI Controller reads EEPROM for initialization
801
Read/write PCI controller registers
802
Boundary scan test for all FPGAs
803
Configure CTRL_FPGA
804
Read/write CTRL_FPGA registers
805
Setup PCI controller for DMA master read/write modes. Transfer data and verify
806
Configure all SEM_FPGAs with a test design, and verify.
807
Hardware ready
808

FIG. 25

```
module register (clock. reset. d. q):
input clock. d. reset;
output q:
reg q:

always@(posedge clock or negedge reset)
     if([illegible]reset)
          q = 0;
     else
          q = d;

endmodule

module example:
     wire d1. d2. d3;
     wire q1. q2. q3;

     reg sigin:
     wire sigout:
     reg clk. reset:

     register reg1 (clk, reset, d1, q1);
     register reg2 (clk, reset. d2, q2);
     register reg3 (clk, reset. d3, q3);

     assign d1 = sigin ^ q3;
     assign d2 = q1 ^ q3;
     assign d3 = q2 ^ q3:
     assign sigout = q3;

     // a clock generator
     always
     begin
          clk = 0:
          #5;
          clk = 1;
          #5:
     end

     // a signal generator
     always
     begin
          #10;
          sigin = $random;
     end

     // initialization
     initial
     begin
          reset = 0;
          sigin = 0;
          #1;
          reset =1;
          #5:
          $monitor($time, " %b, %b", sigin, sigout);
          #1000 $finish;
     end
     end module
```

FIG. 26

CIRCUIT DIAGRAM

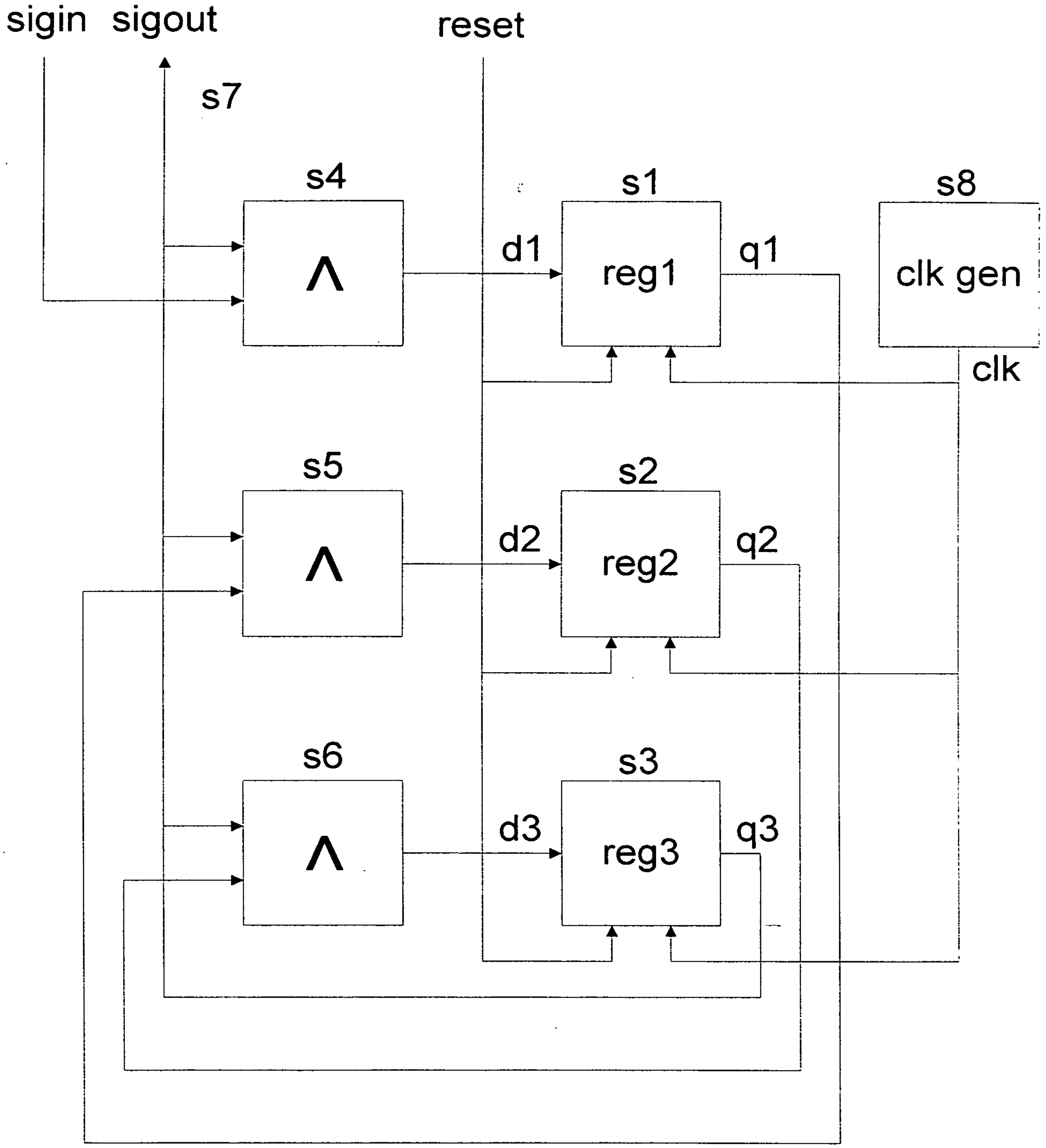
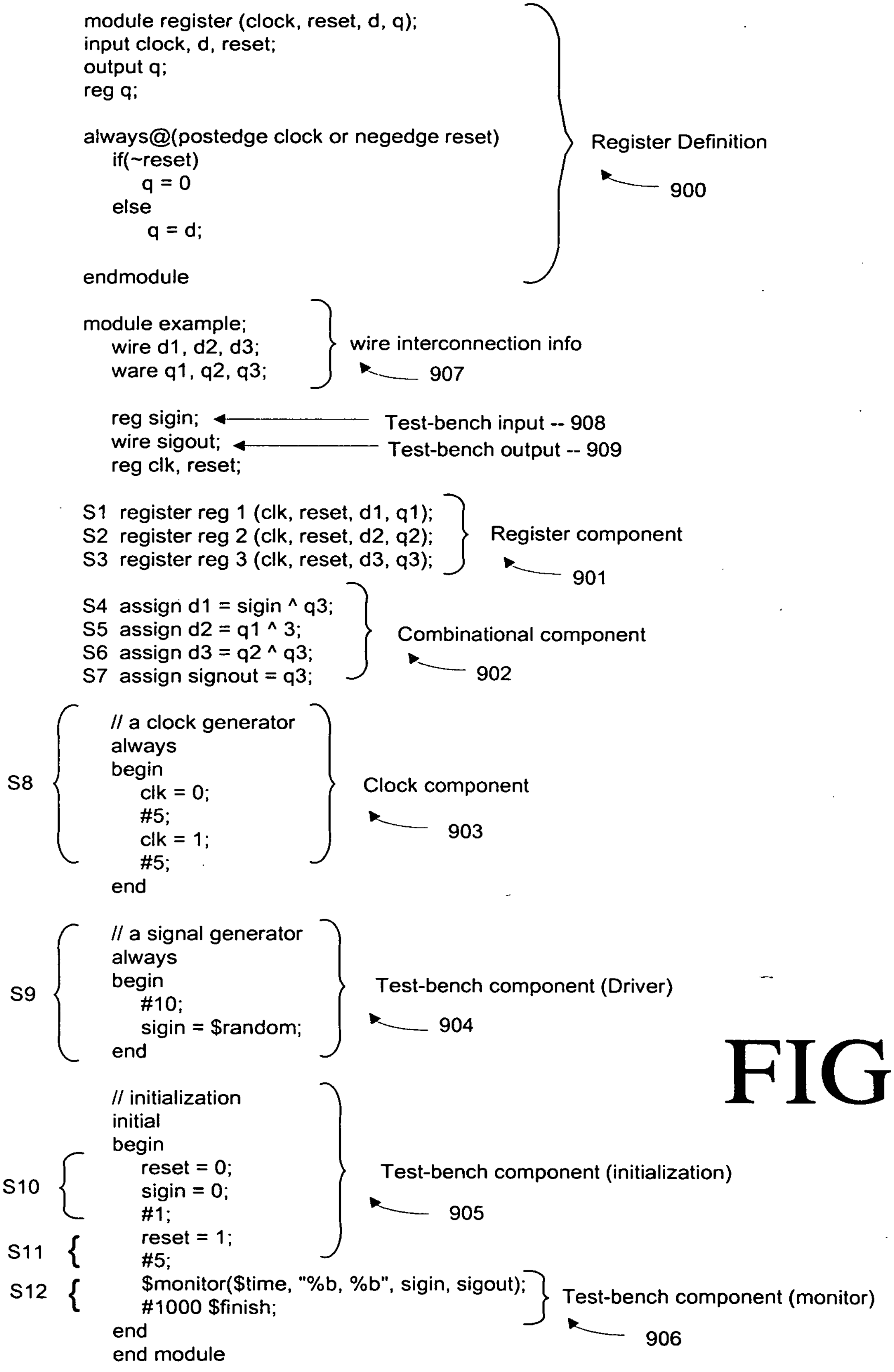


FIG. 27



```
    module register (clock, reset, d, q);
    input clock, d, reset;
    output q;
    reg q;

    always@(postedge clock or negedge reset)
        if(~reset)
            q = 0
        else
             q = d;

    endmodule

    module example;
        wire d1, d2, d3;
        ware q1, q2, q3;

        reg sigin;
        wire sigout;
        reg clk, reset;

    S1  register reg 1 (clk, reset, d1, q1);
    S2  register reg 2 (clk, reset, d2, q2);
    S3  register reg 3 (clk, reset, d3, q3);

    S4  assign d1 = sigin ^ q3;
    S5  assign d2 = q1 ^ 3;
    S6  assign d3 = q2 ^ q3;
    S7  assign signout = q3;

        // a clock generator
        always
        begin
S8          clk = 0;
            #5;
            clk = 1;
            #5;
        end

        // a signal generator
        always
        begin
S9          #10;
            sigin = $random;
        end

        // initialization
        initial
        begin
            reset = 0;
S10         sigin = 0;
            #1;
S11         reset = 1;
            #5;
S12         $monitor($time, "%b, %b", sigin, sigout);
            #1000 $finish;
        end
        end module
```

FIG. 28

SIGNAL NETWORK ANALYSIS

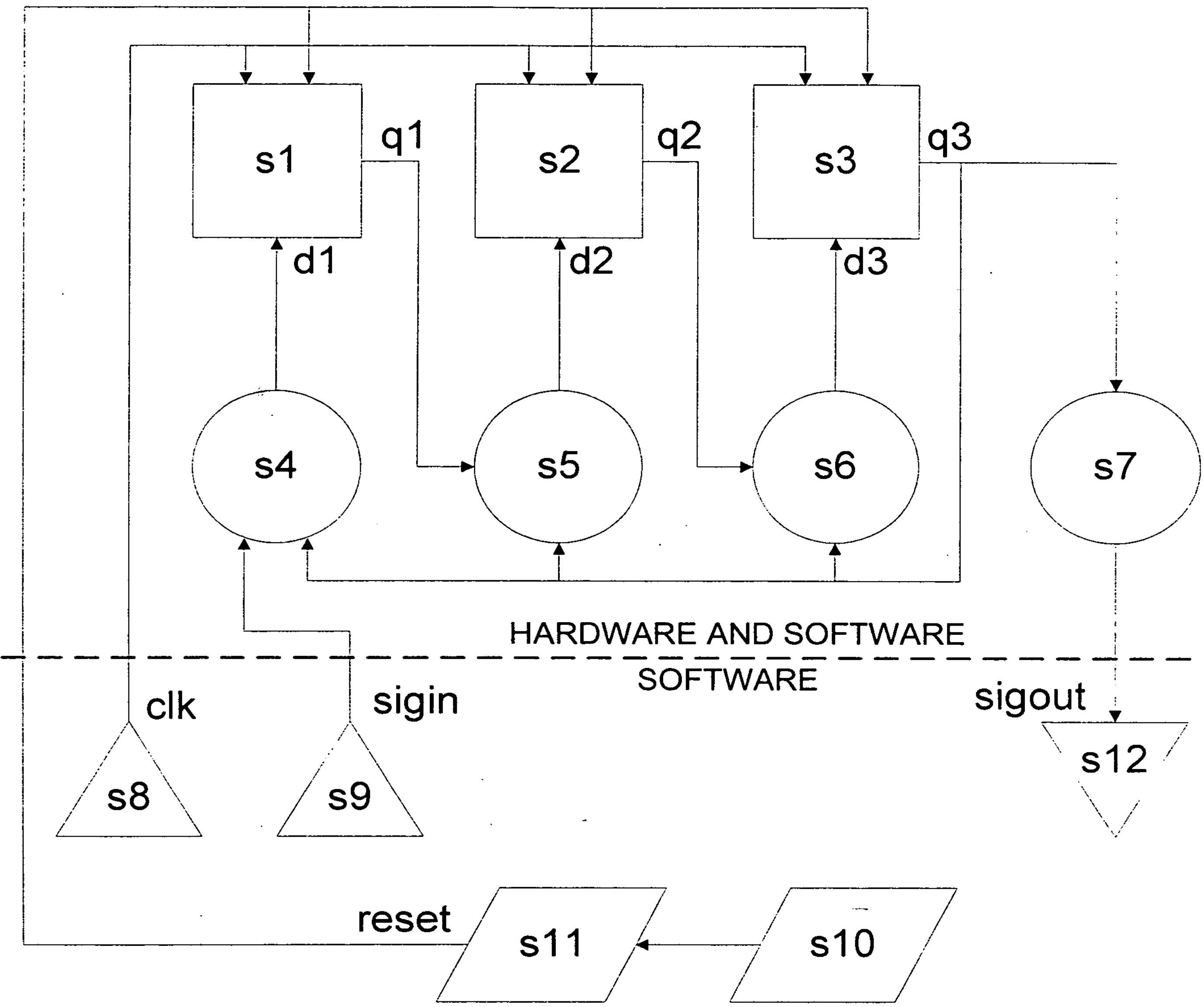


FIG. 29

SOFTWARE/HARDWARE PARTITION RESULT

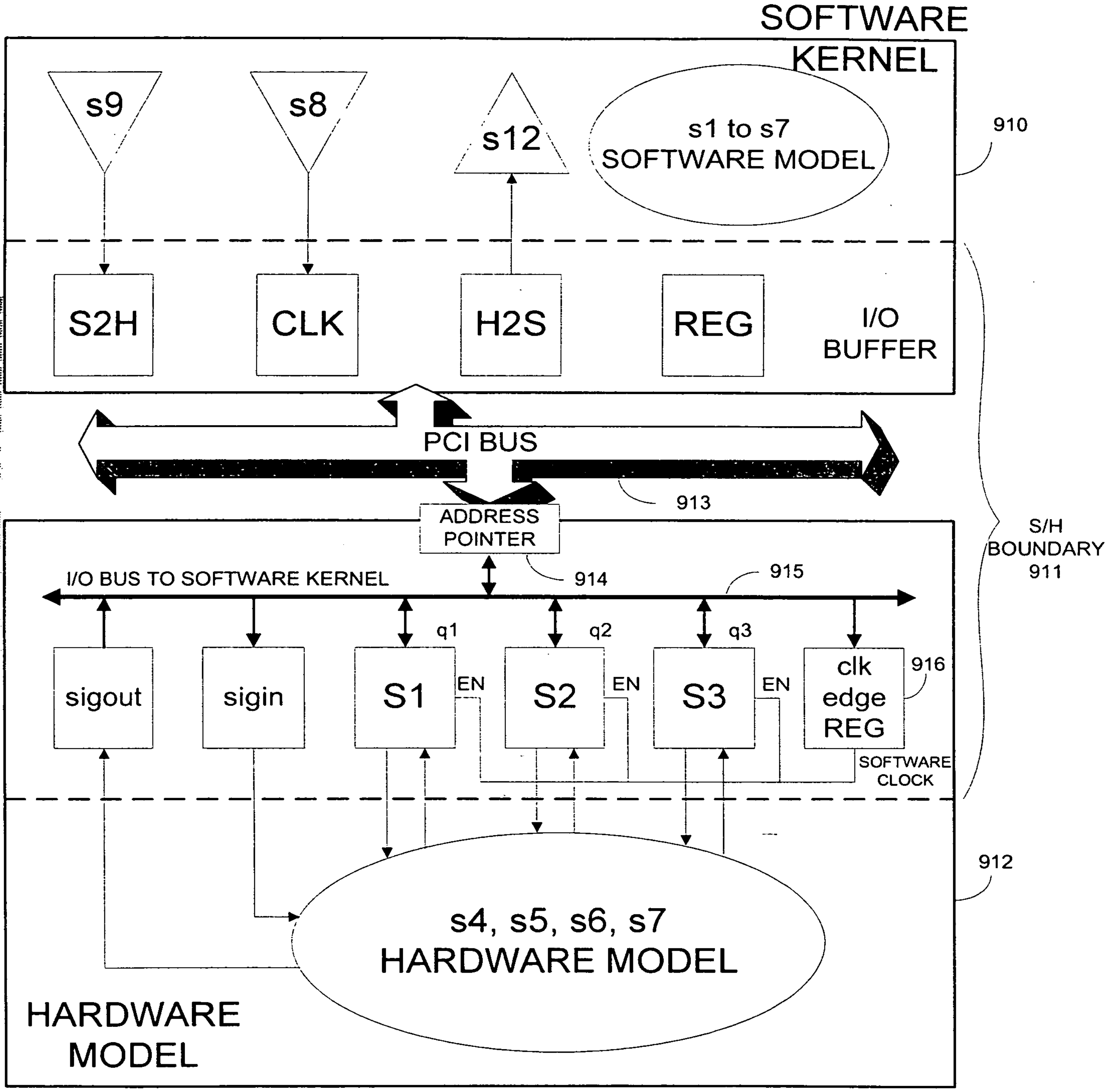


FIG. 30

HARDWARE MODEL

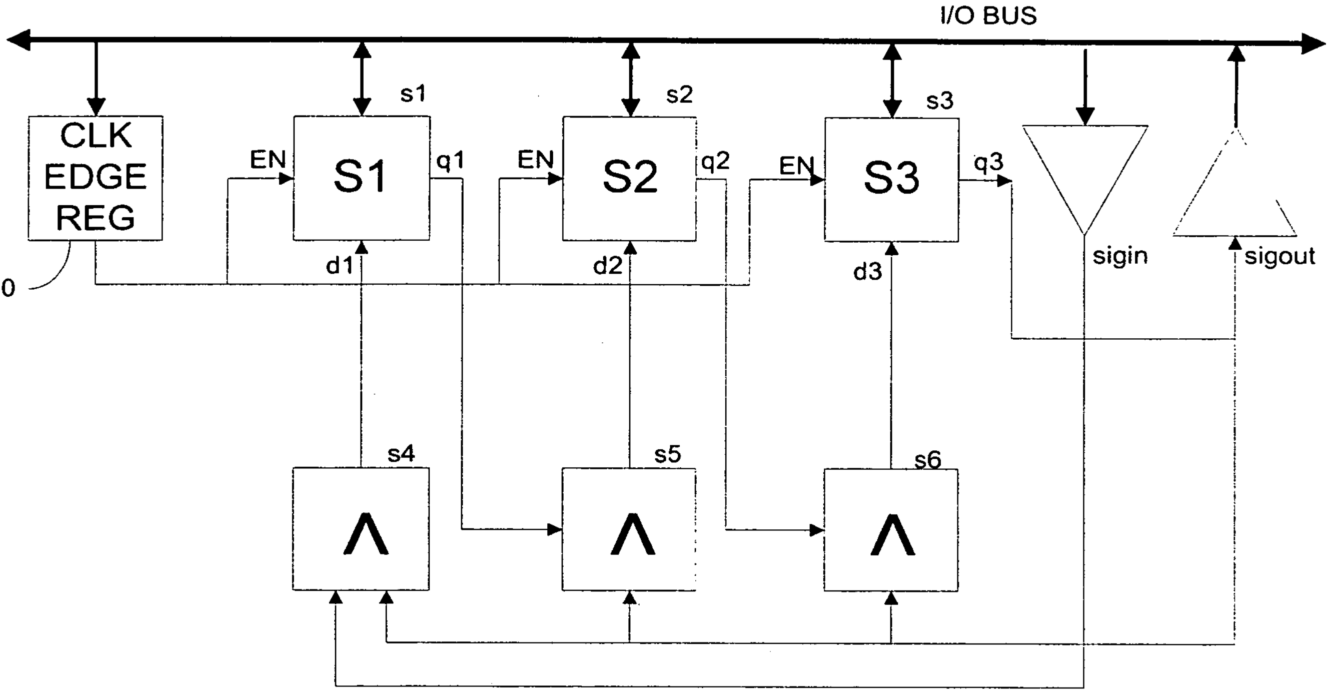
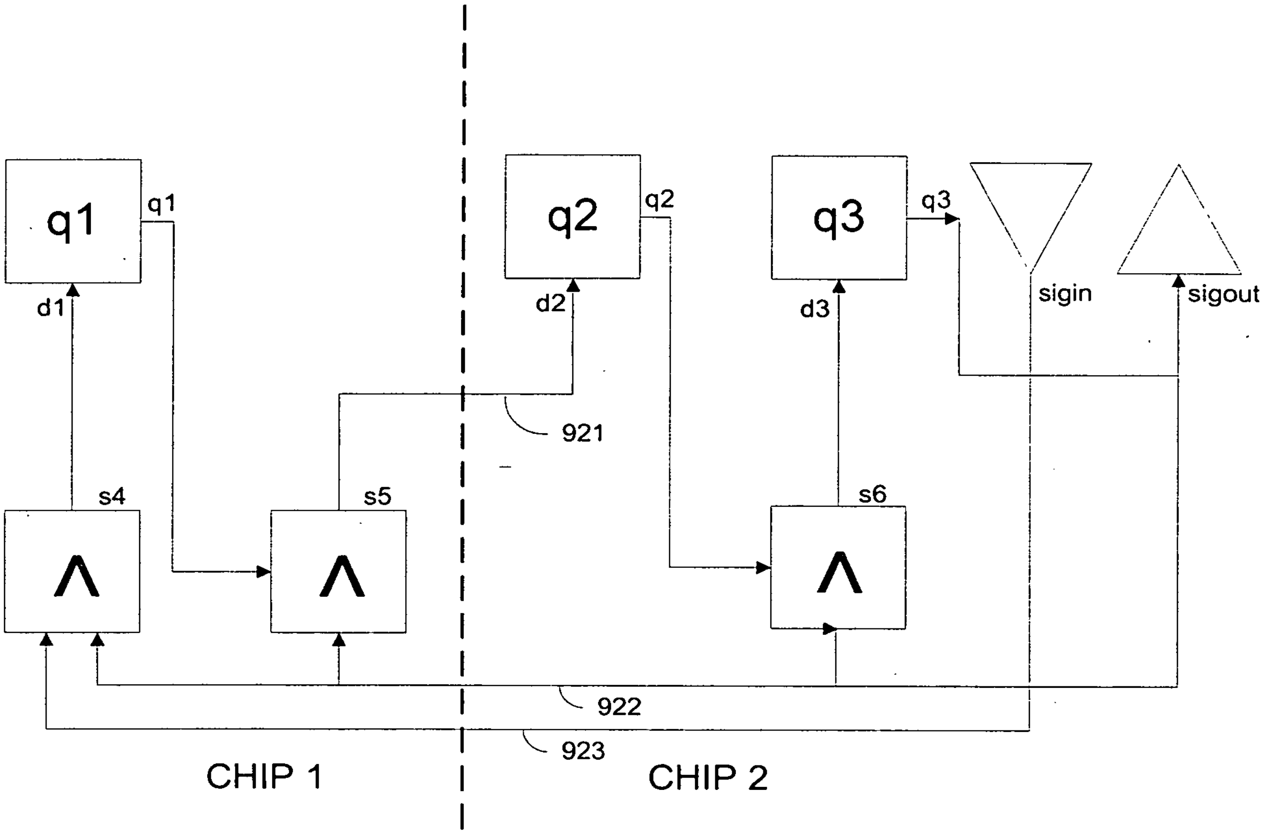


FIG. 31

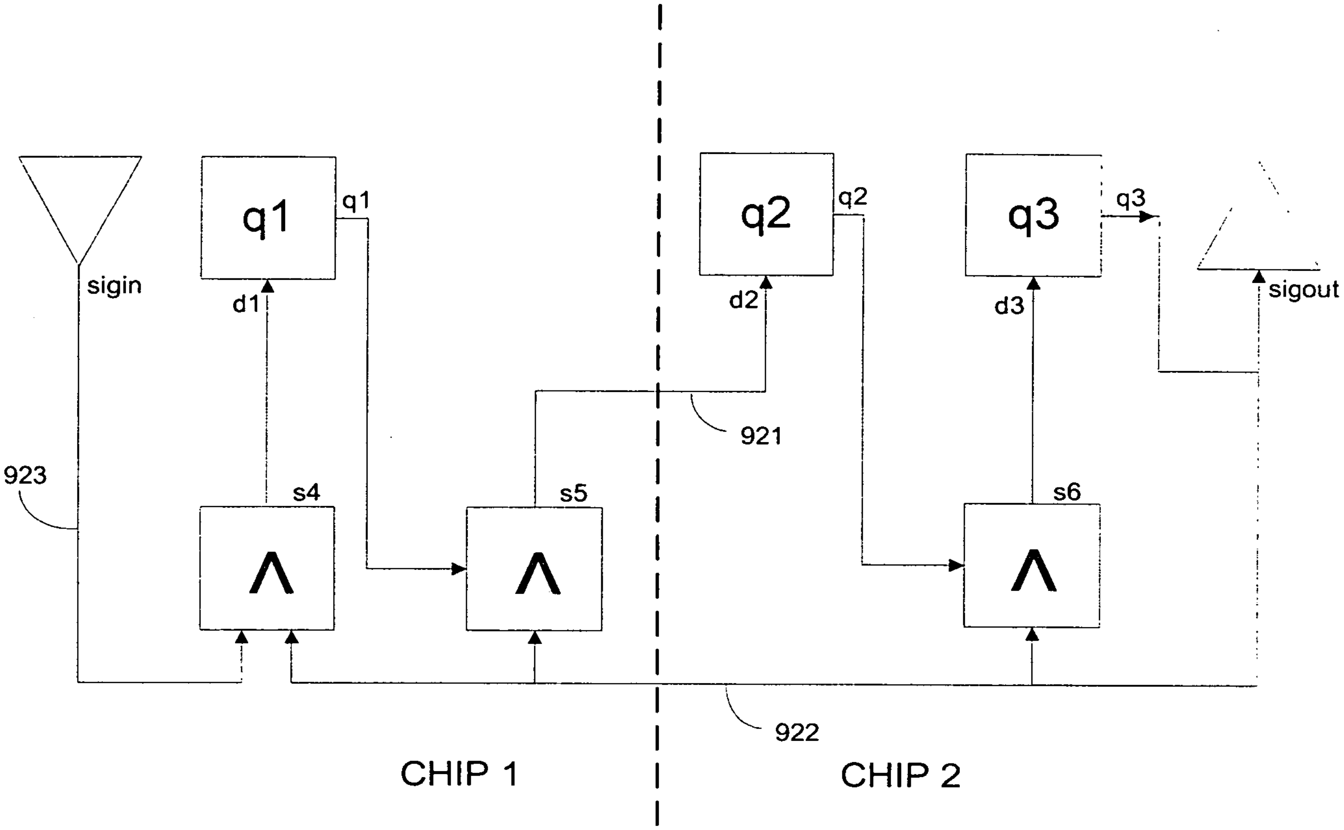
PARTITION RESULT #1



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

PARTITION RESULT #2



(IGNORE I/O AND CLOCK EDGE REGISTER).

FIG. 33

LOGIC PATCHING

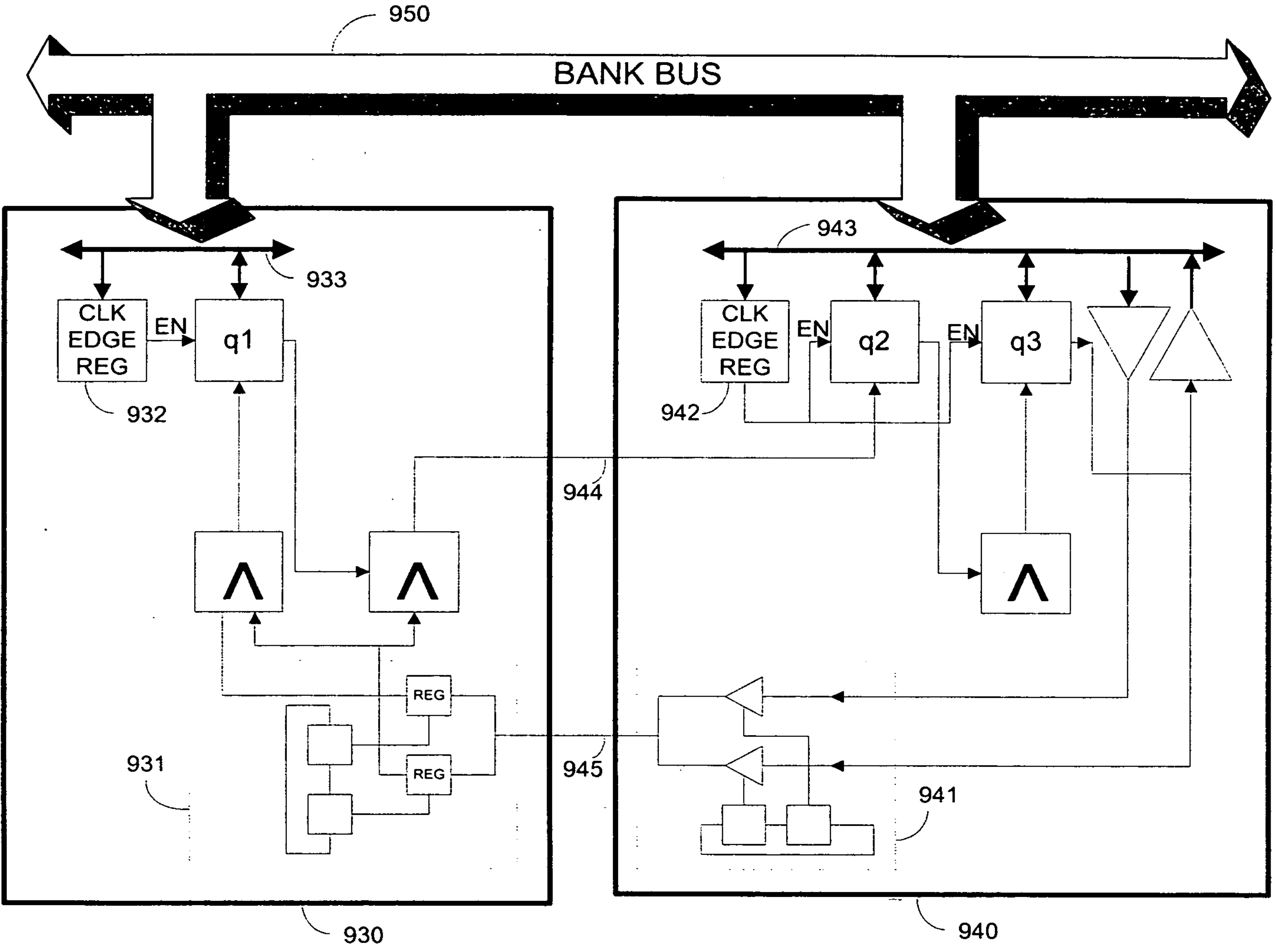


FIG. 34

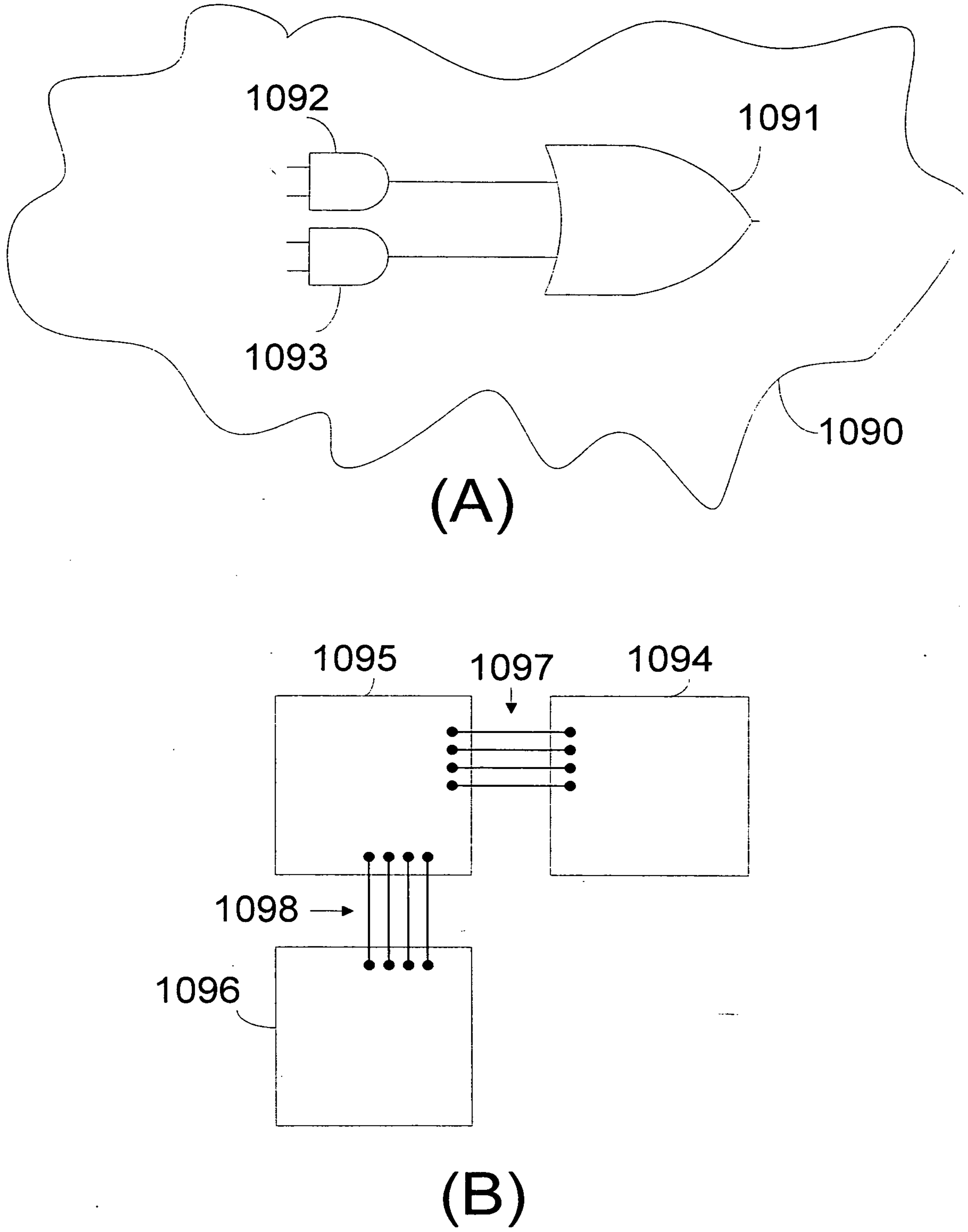
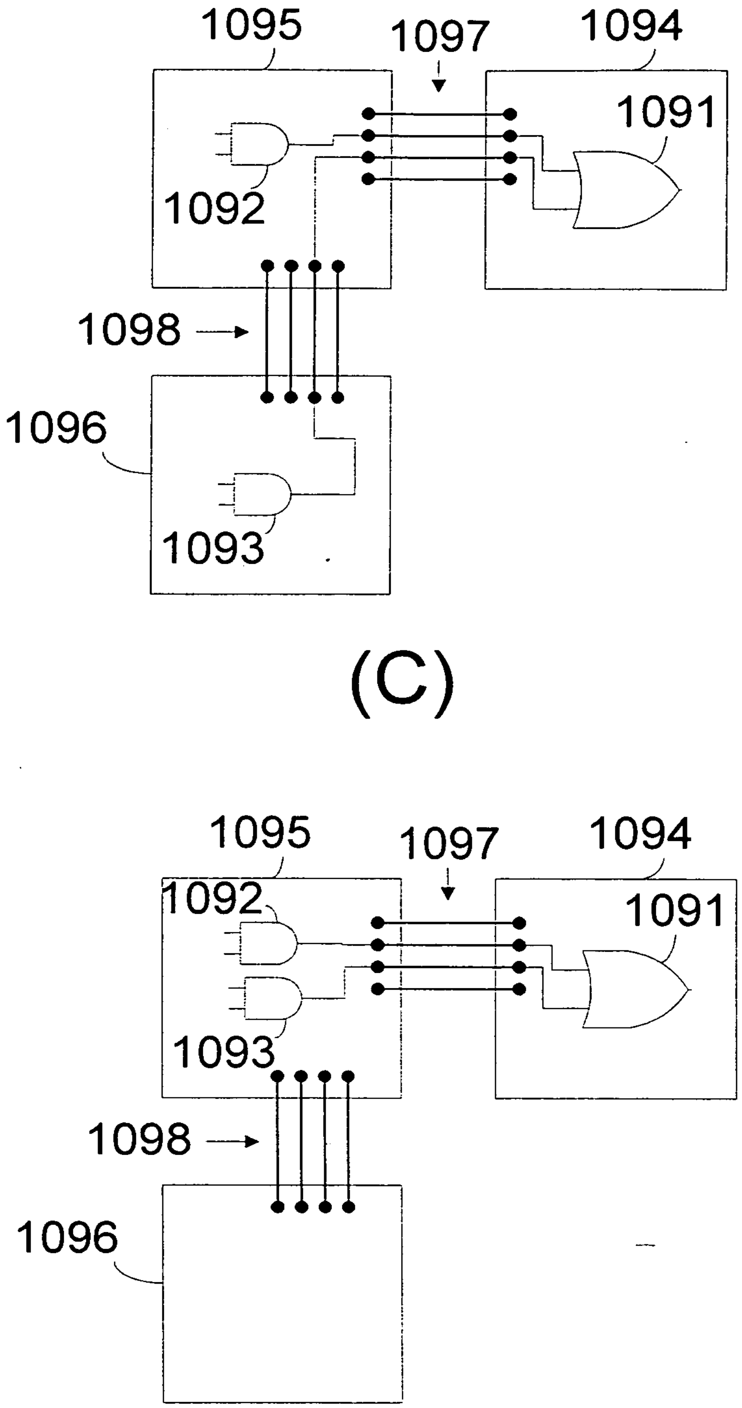


FIG. 35


1095
1097
1094
1091
1092
1098
1096
1093
1095
1097
1094
1092
1091
1093
1098
1096

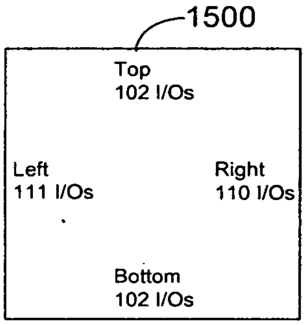
(C)

(D)

FIG. 35

I/O PIN OVERVIEW OF FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V with 599-pin PGA package



425 Interconnect I/O pins

45 Dedicated I/O pins:

GCLK, FD_BUS[31..0], F_RD, F_WR,
DATAXSFR, SHIFTIN, SHIFTOUT,
SPACE[2..0], EVAL, EV_REQ_N,
DEV_OE, DEV_CLRN

FIG. 36

FPGA INTERCONNECT BUSES

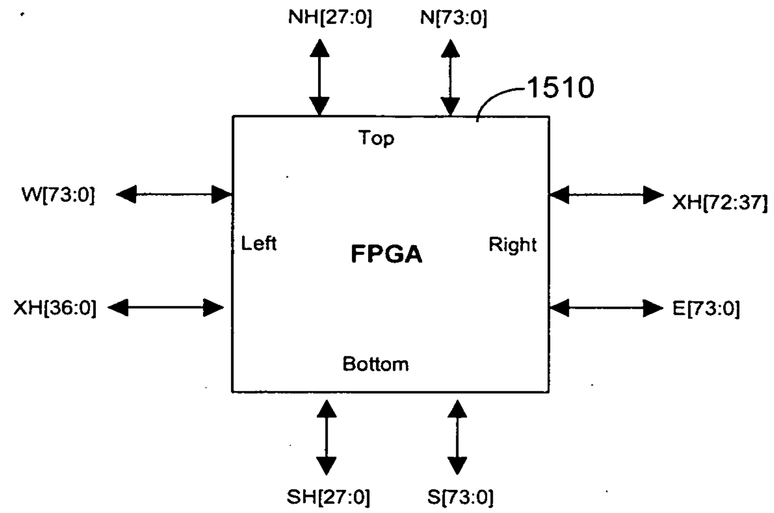


FIG. 37

BOARD CONNECTION - SIDE VIEW

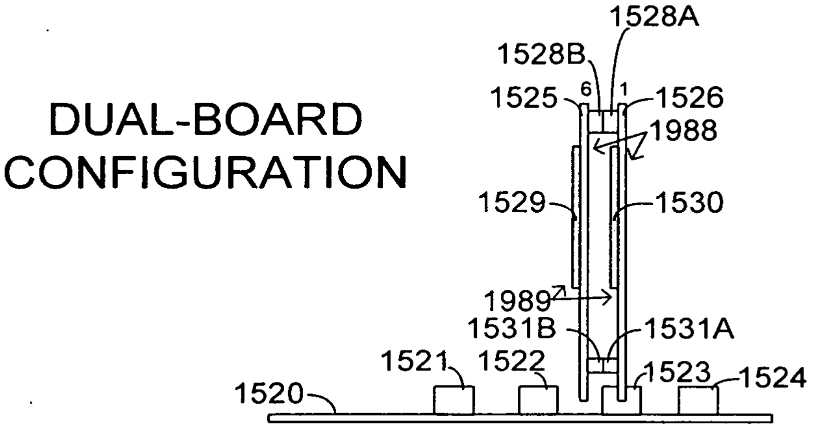


FIG. 38(A)

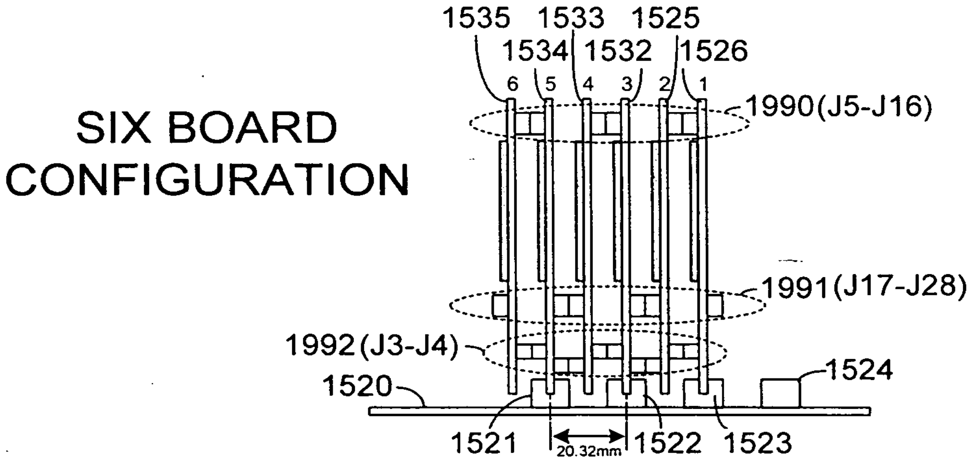


FIG. 38(B)

SIX-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

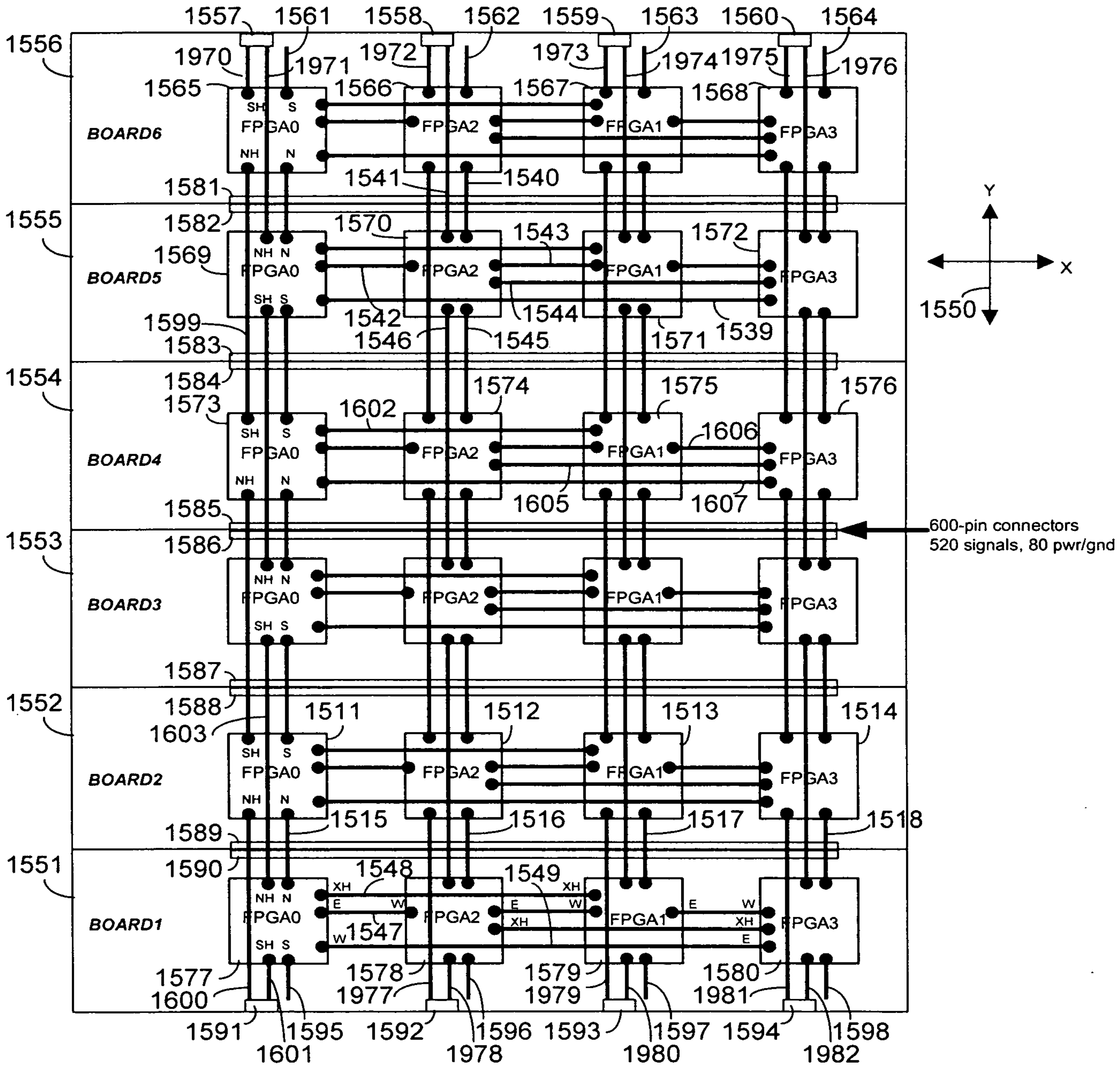


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

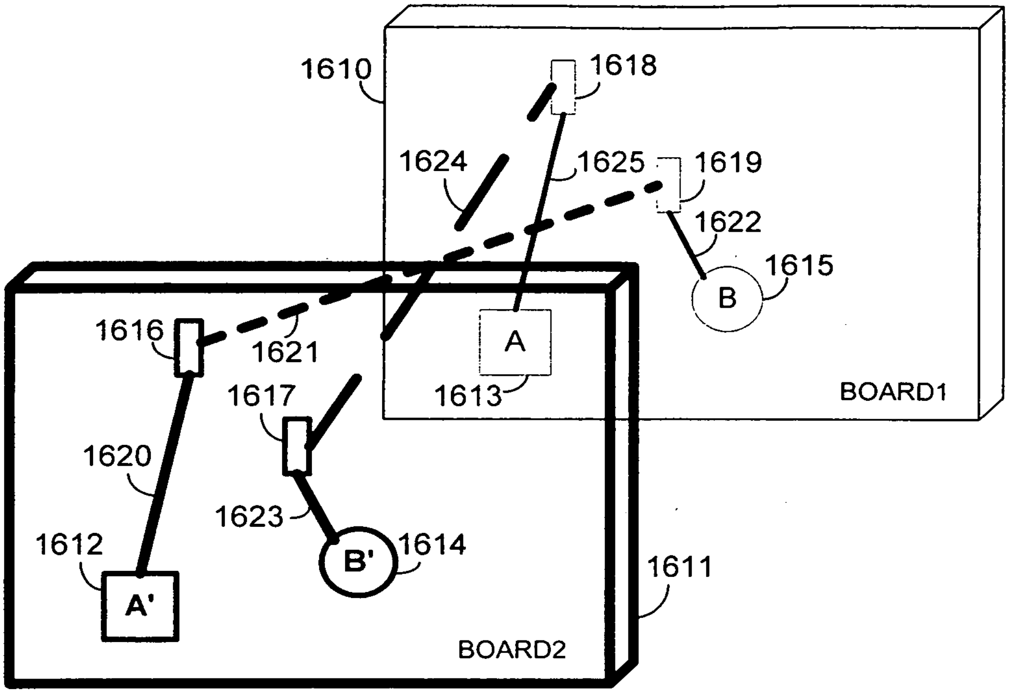


FIG. 40(A)

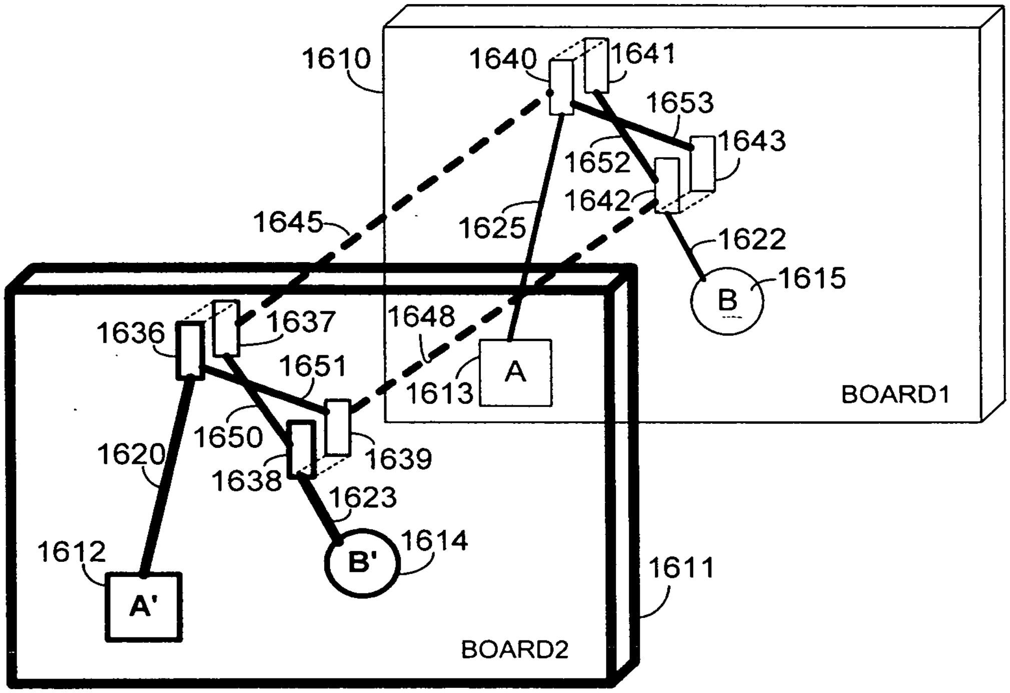


FIG. 40(B)

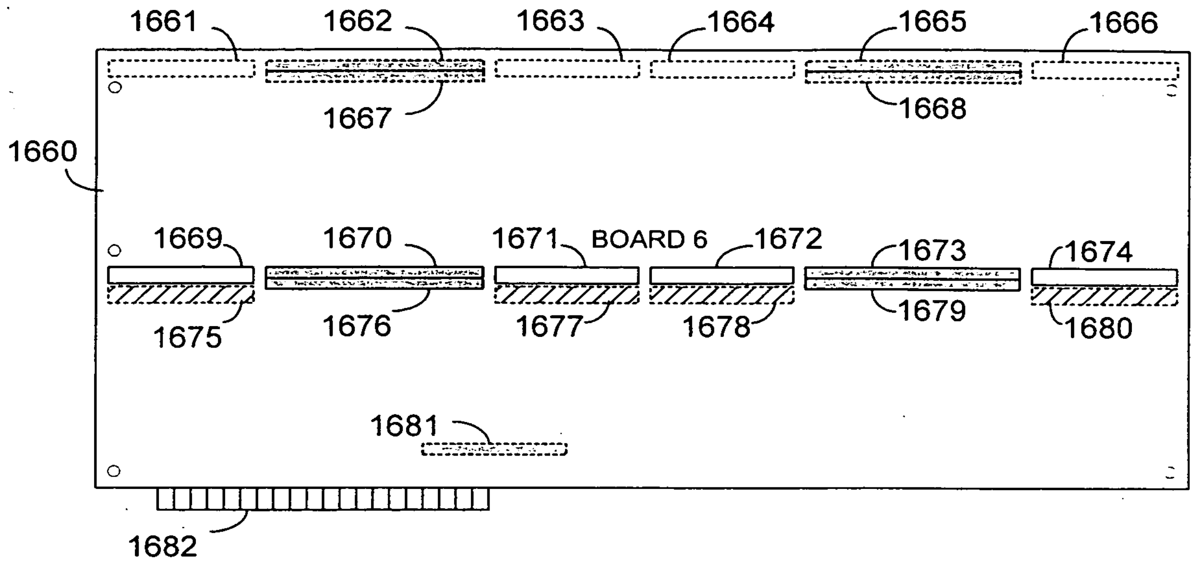


FIG. 41(A)

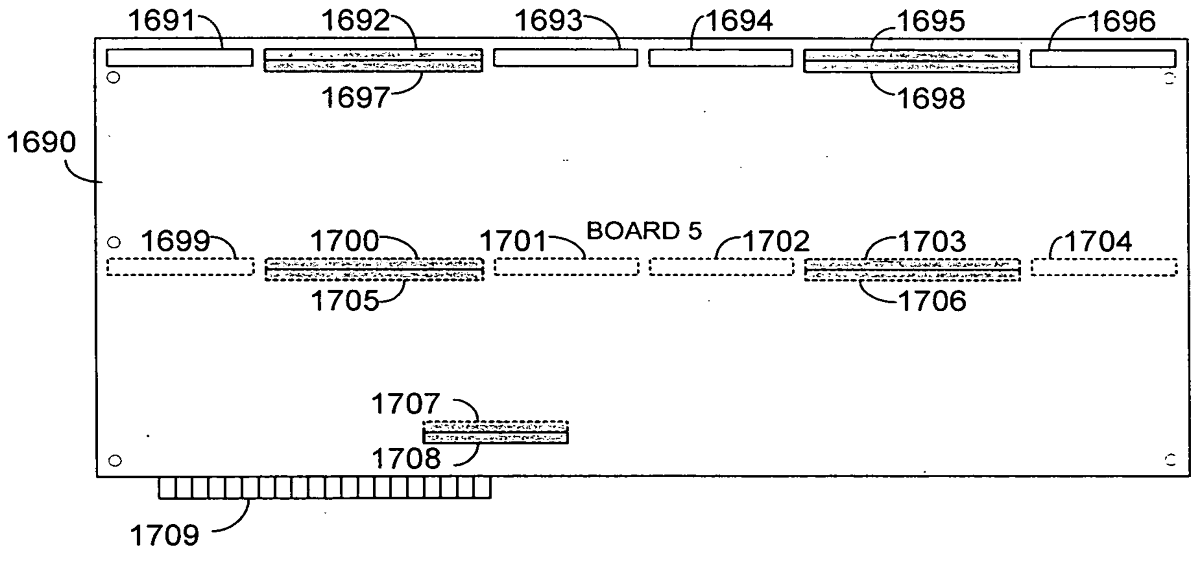


FIG. 41(B)

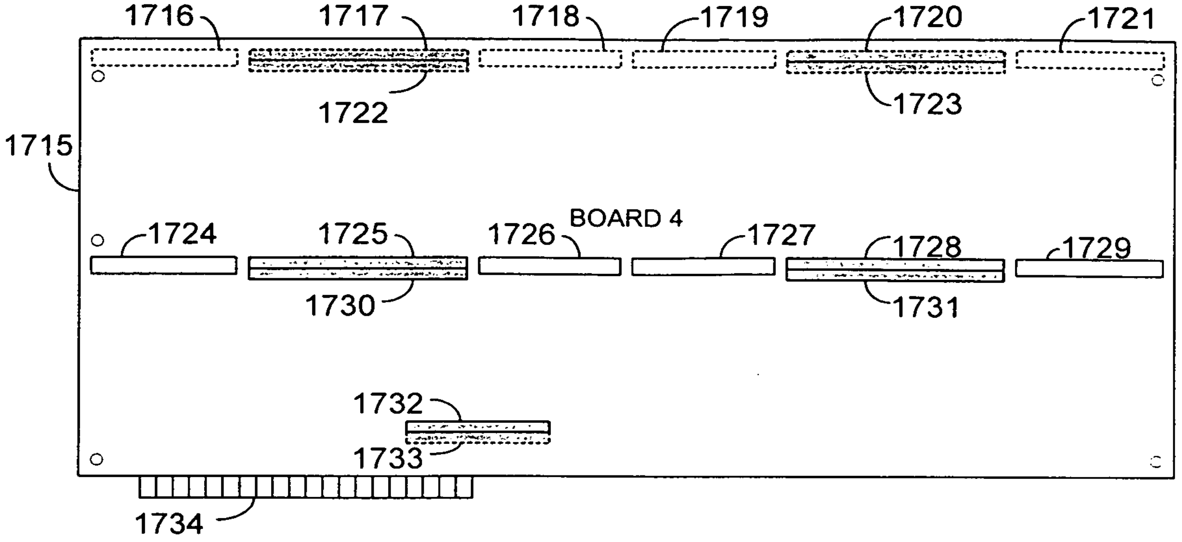


FIG. 41(C)

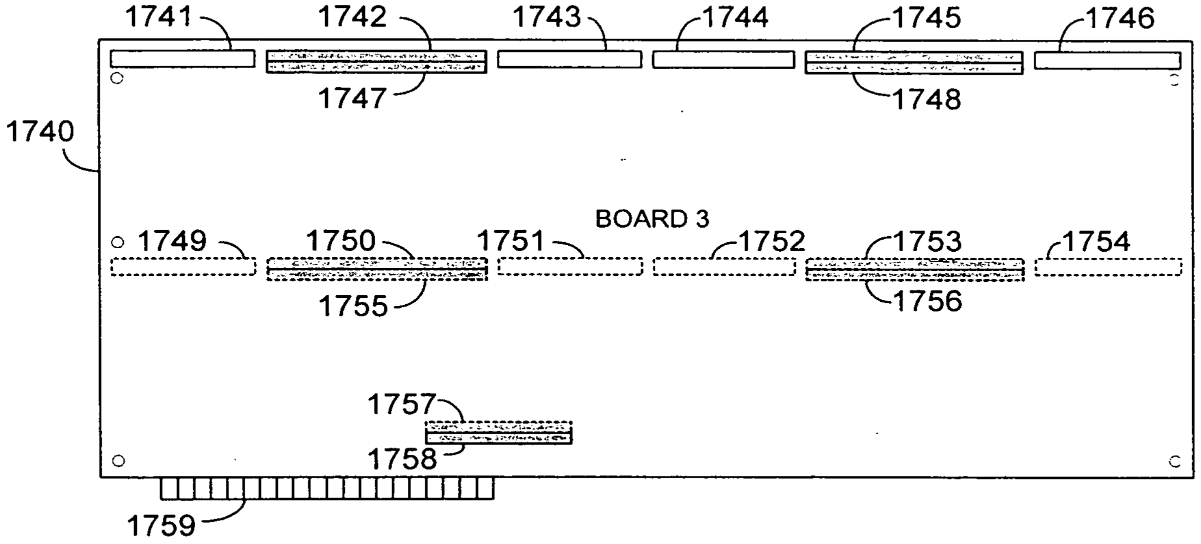


FIG. 41(D)

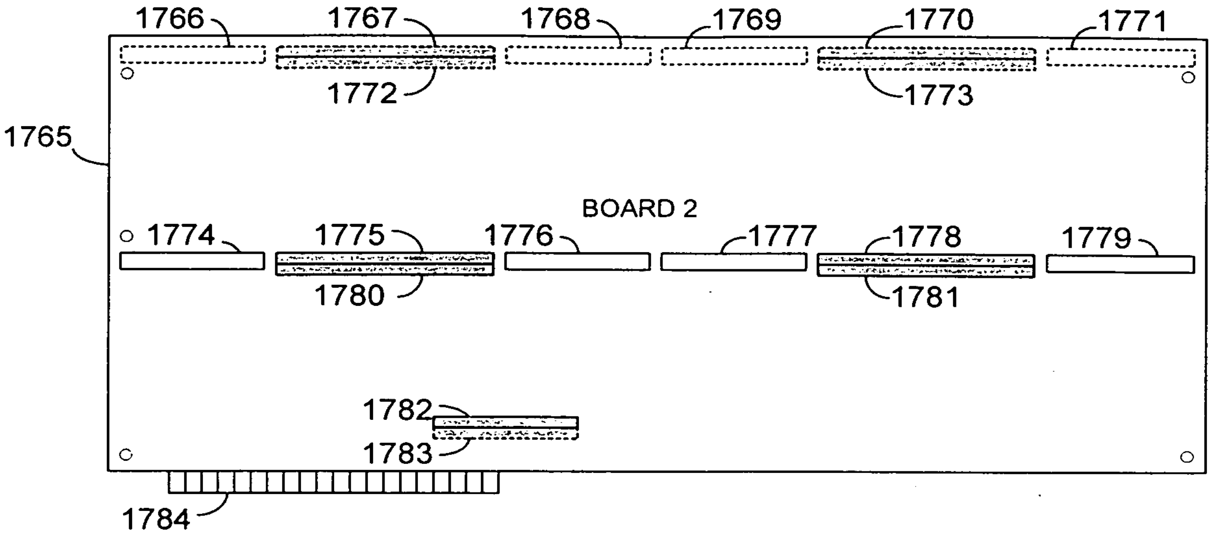


FIG. 41(E)

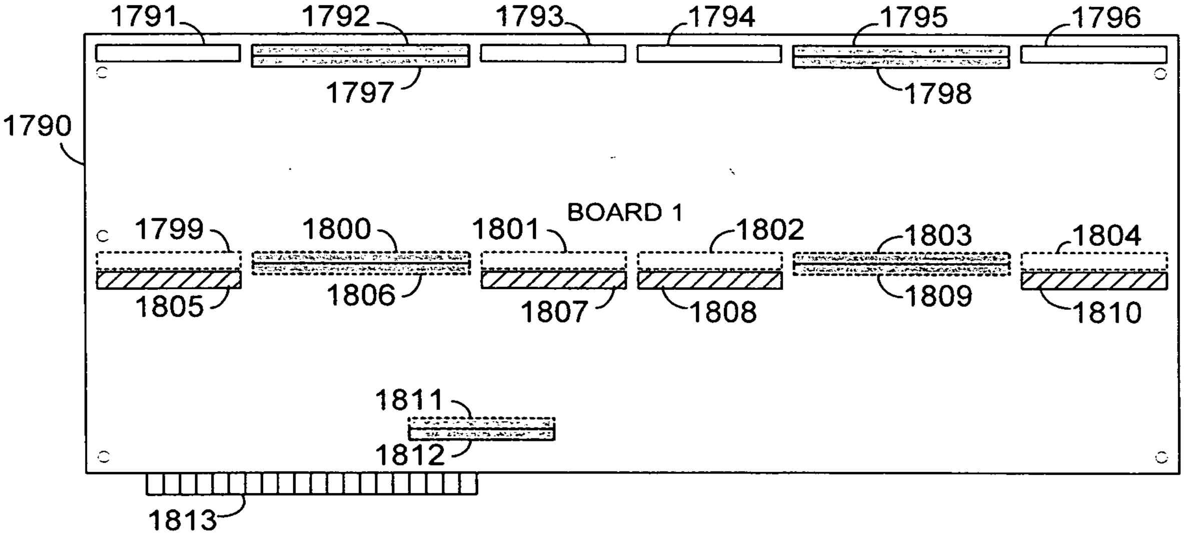


FIG. 41(F)

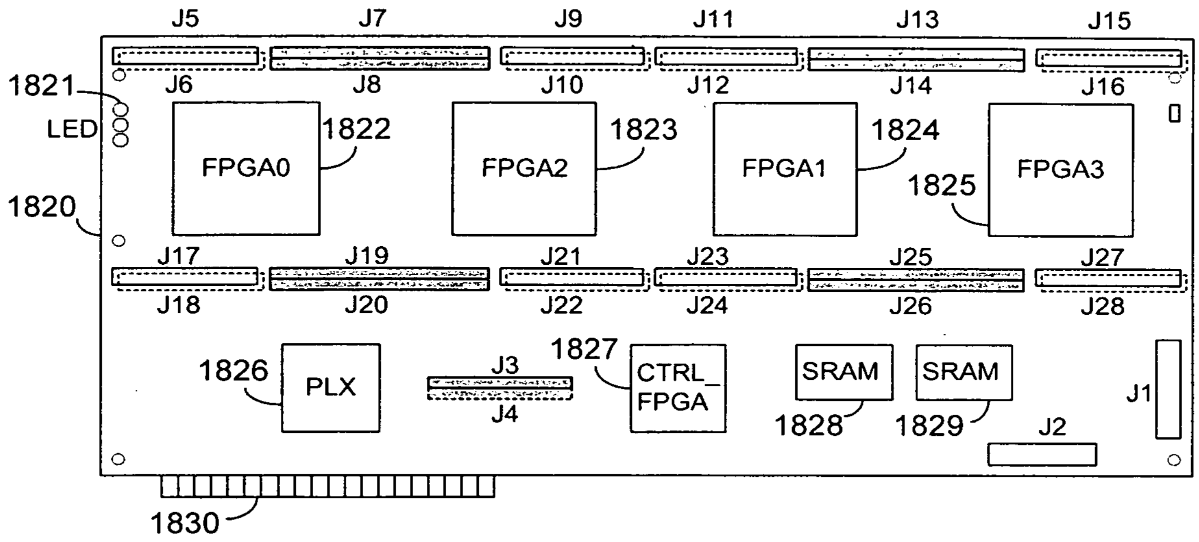

J5
J7
J9
J11
J13
J15
J6
J8
J10
J12
J14
J16
1821
LED
FPGA0
1822
FPGA2
1823
FPGA1
1824
FPGA3
1825
1820
J17
J19
J21
J23
J25
J27
J18
J20
J22
J24
J26
J28
1826
PLX
J3
J4
1827
CTRL_
FPGA
SRAM
1828
SRAM
1829
J1
J2
1830

FIG. 42

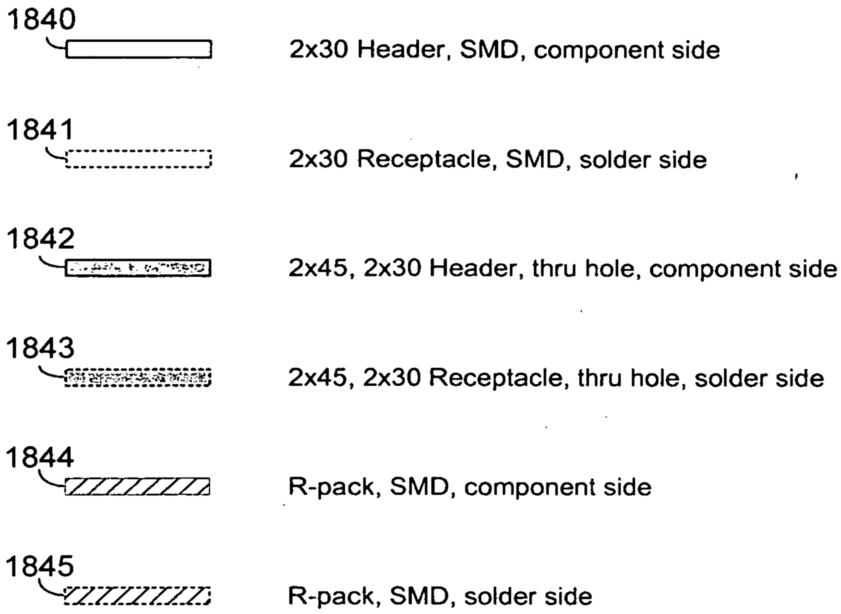

1840
2x30 Header, SMD, component side
1841
2x30 Receptacle, SMD, solder side
1842
2x45, 2x30 Header, thru hole, component side
1843
2x45, 2x30 Receptacle, thru hole, solder side
1844
R-pack, SMD, component side
1845
R-pack, SMD, solder side

FIG. 43

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

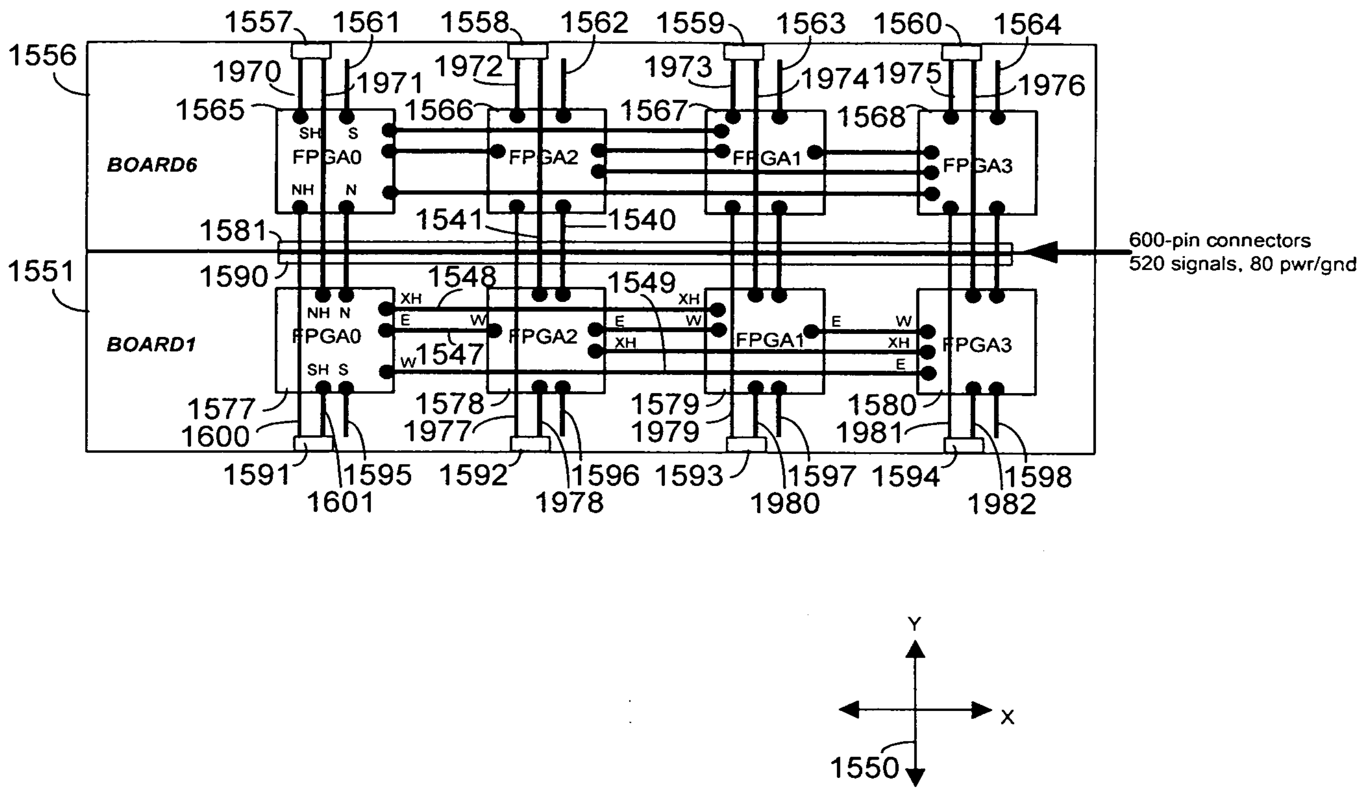


FIG. 44

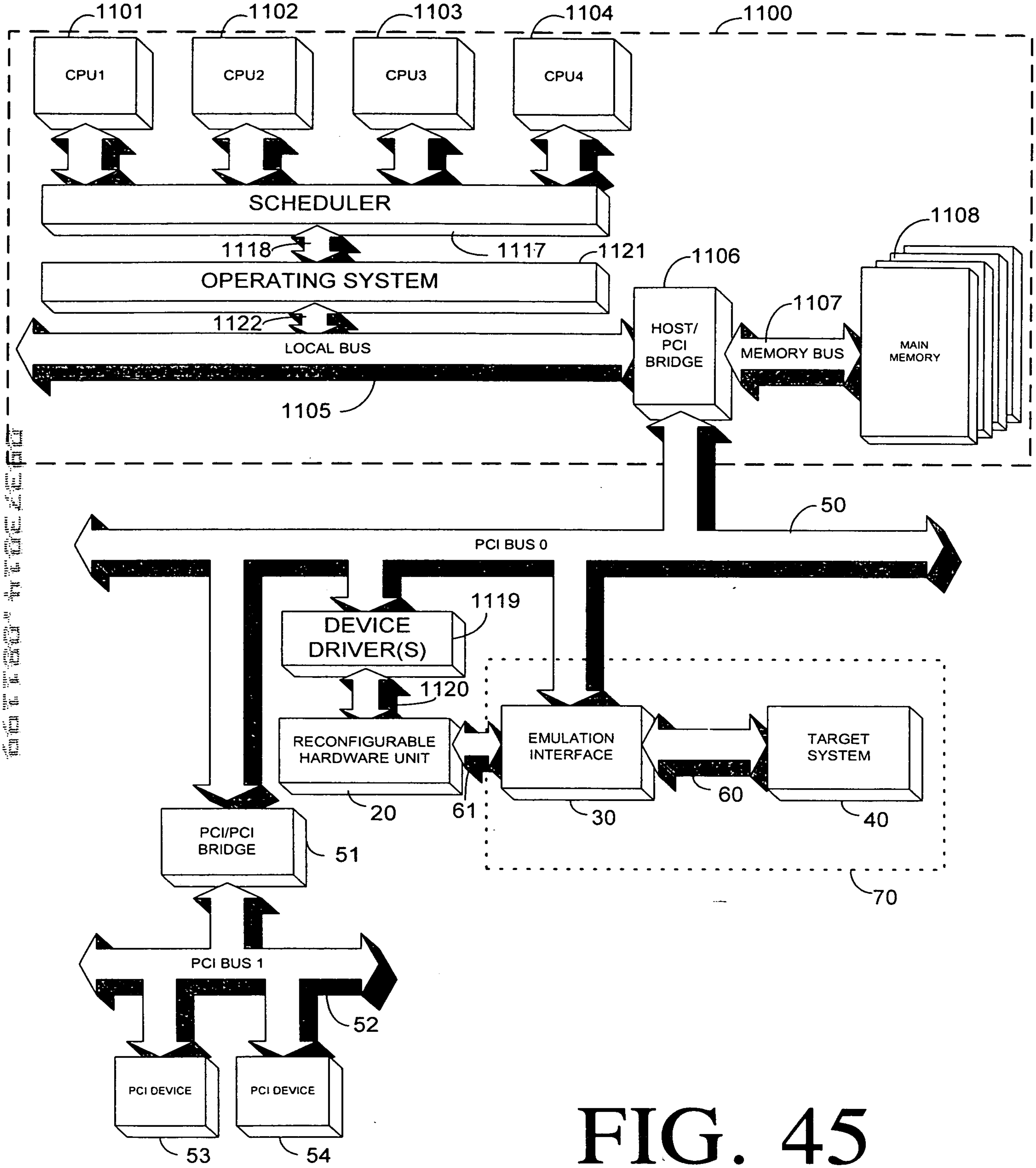

1101
1102
1103
1104
1100
CPU1
CPU2
CPU3
CPU4
SCHEDULER
1118
1117
1121
OPERATING SYSTEM
1106
1108
1122
LOCAL BUS
HOST/
PCI
BRIDGE
1107
MEMORY BUS
MAIN
MEMORY
1105
50
PCI BUS 0
1119
DEVICE
DRIVER(S)
1120
RECONFIGURABLE
HARDWARE UNIT
EMULATION
INTERFACE
TARGET
SYSTEM
20
61
30
60
40
PCI/PCI
BRIDGE
51
70
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG. 45

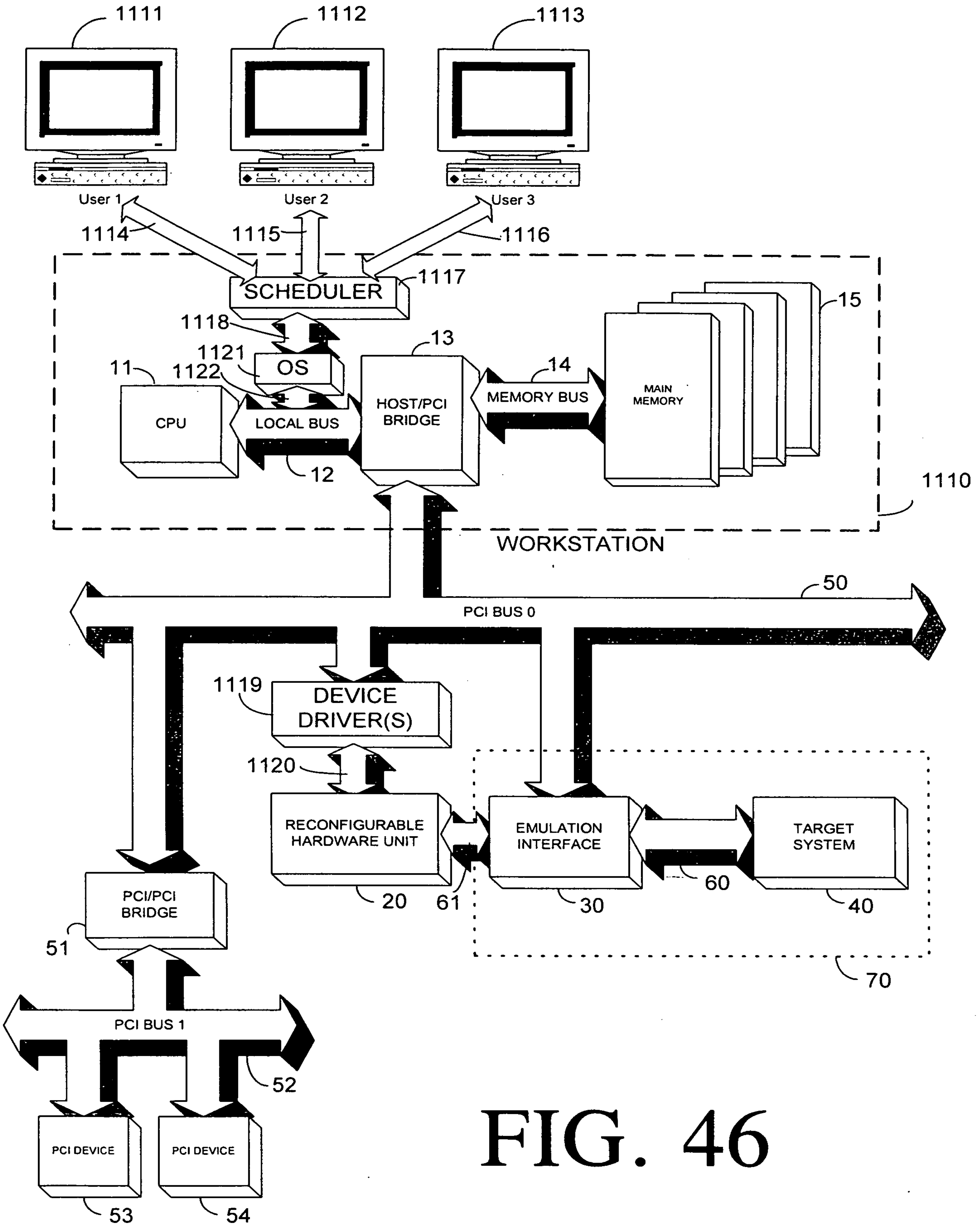

1111
1112
1113
User 1
User 2
User 3
1114
1115
1116
SCHEDULER
1117
1118
1121
OS
1122
11
CPU
LOCAL BUS
12
13
HOST/PCI BRIDGE
14
MEMORY BUS
MAIN MEMORY
15
1110
WORKSTATION
50
PCI BUS 0
1119
DEVICE DRIVER(S)
1120
RECONFIGURABLE HARDWARE UNIT
20
61
EMULATION INTERFACE
30
60
TARGET SYSTEM
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
53
PCI DEVICE
54

FIG. 46

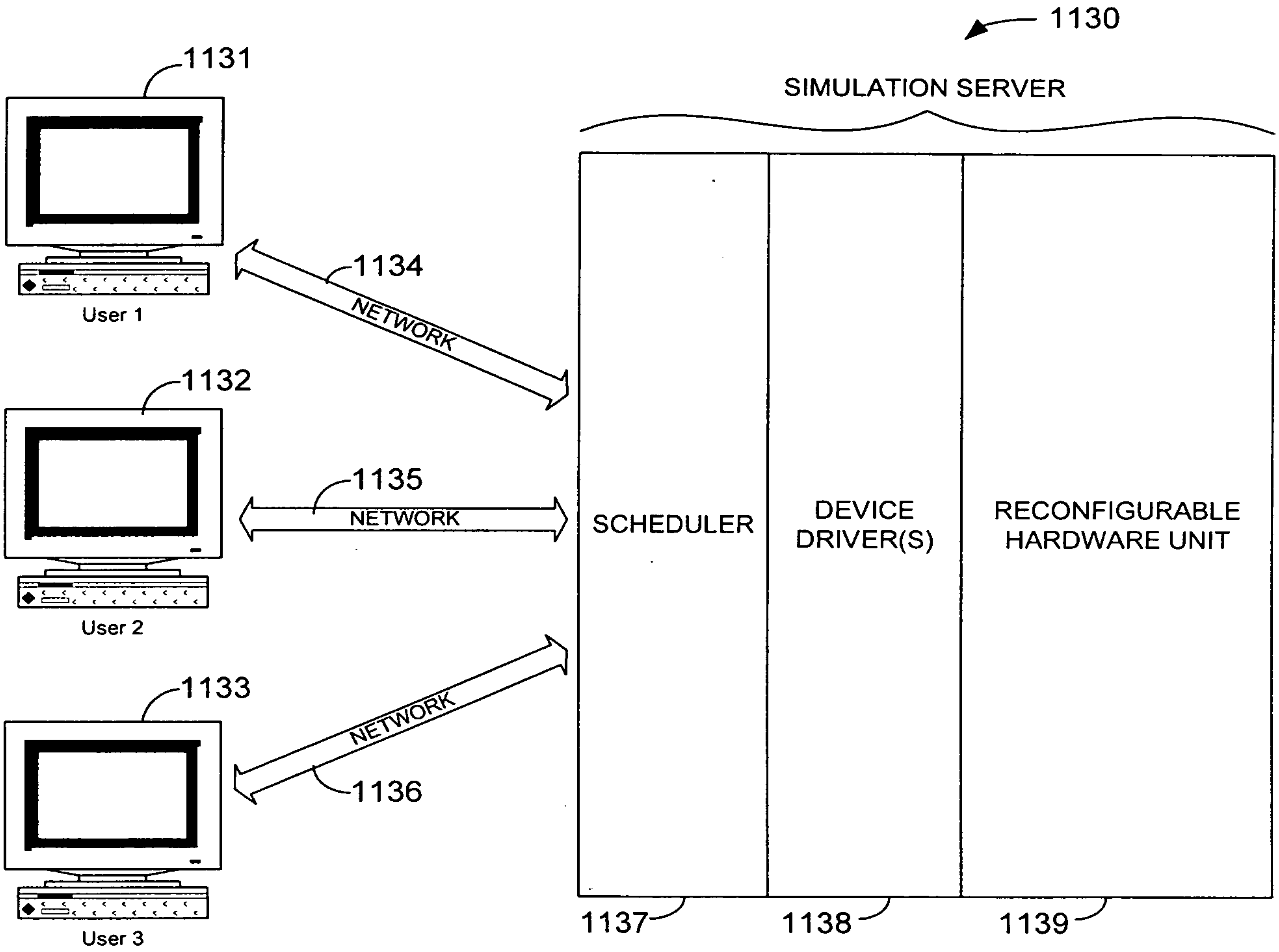

1130
SIMULATION SERVER
1131
User 1
1134
NETWORK
1132
User 2
1135
NETWORK
1133
User 3
NETWORK
1136
SCHEDULER
DEVICE DRIVER(S)
RECONFIGURABLE HARDWARE UNIT
1137
1138
1139

FIG. 47

SIMULATION SERVER ARCHITECTURE

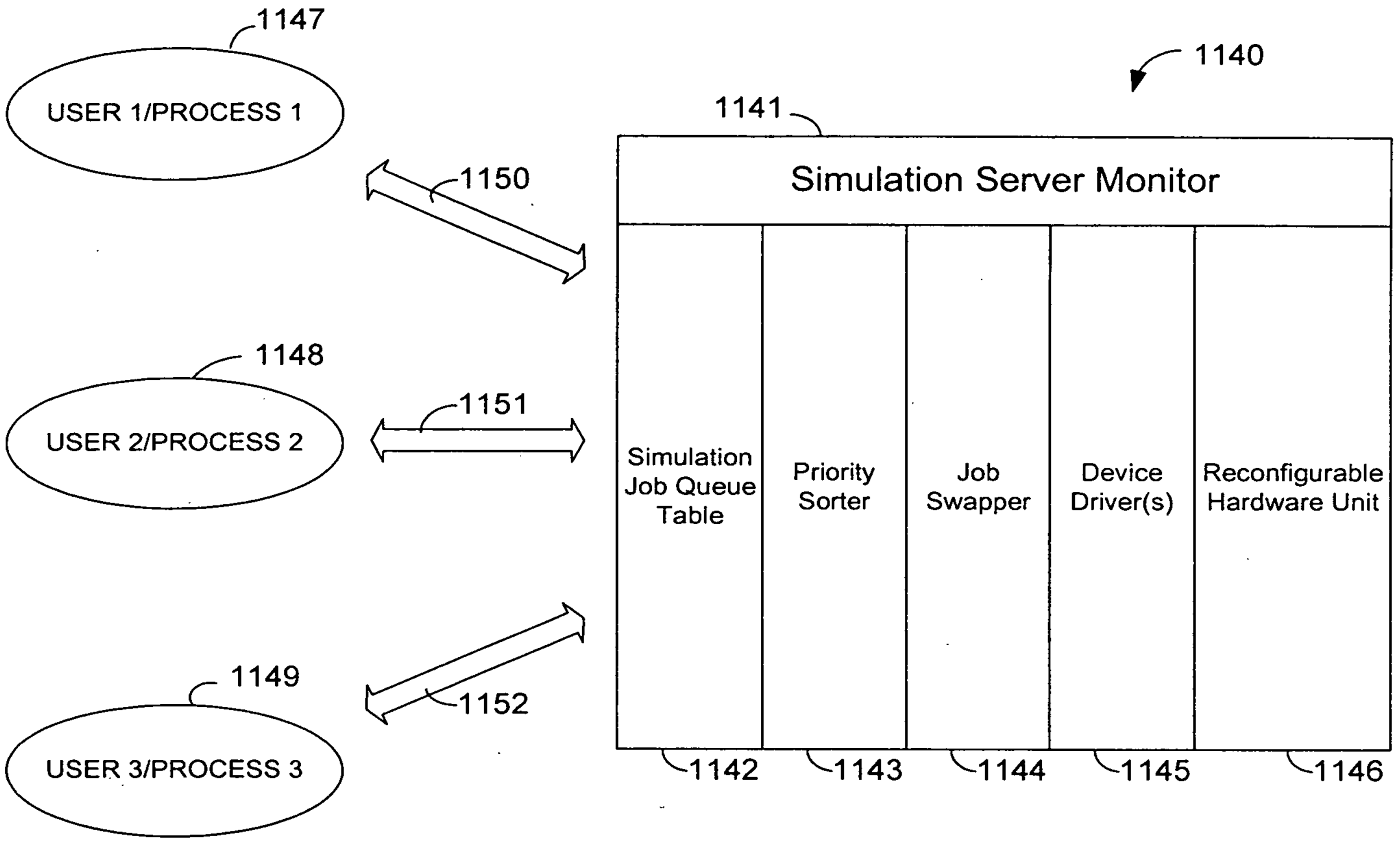


FIG. 48

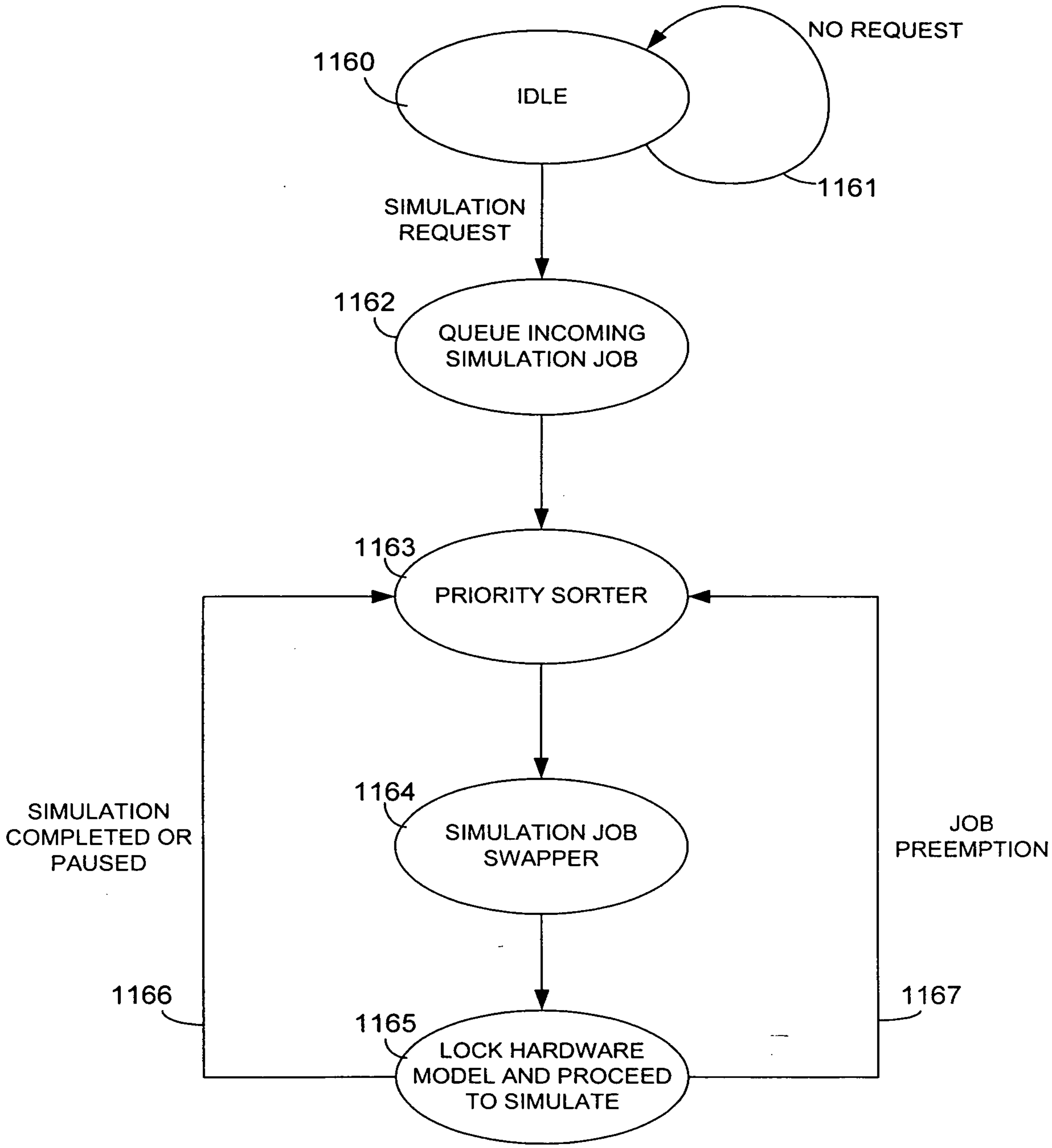

1160
IDLE
NO REQUEST
1161
SIMULATION REQUEST
1162
QUEUE INCOMING SIMULATION JOB
1163
PRIORITY SORTER
SIMULATION COMPLETED OR PAUSED
1164
SIMULATION JOB SWAPPER
JOB PREEMPTION
1166
1167
1165
LOCK HARDWARE MODEL AND PROCEED TO SIMULATE

FIG. 49

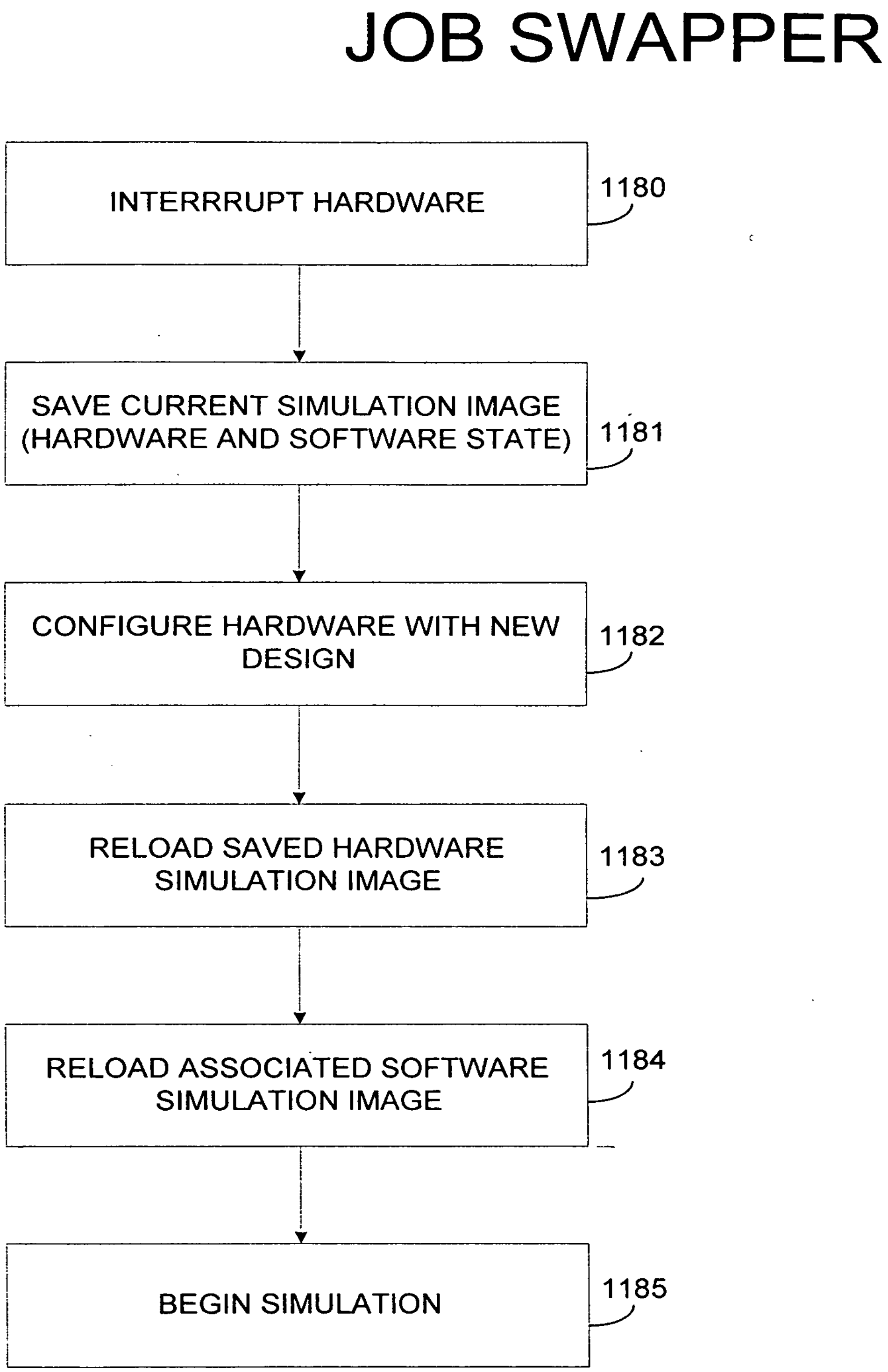

JOB SWAPPER
INTERRRUPT HARDWARE
1180
SAVE CURRENT SIMULATION IMAGE (HARDWARE AND SOFTWARE STATE)
1181
CONFIGURE HARDWARE WITH NEW DESIGN
1182
RELOAD SAVED HARDWARE SIMULATION IMAGE
1183
RELOAD ASSOCIATED SOFTWARE SIMULATION IMAGE
1184
BEGIN SIMULATION
1185

FIG. 50

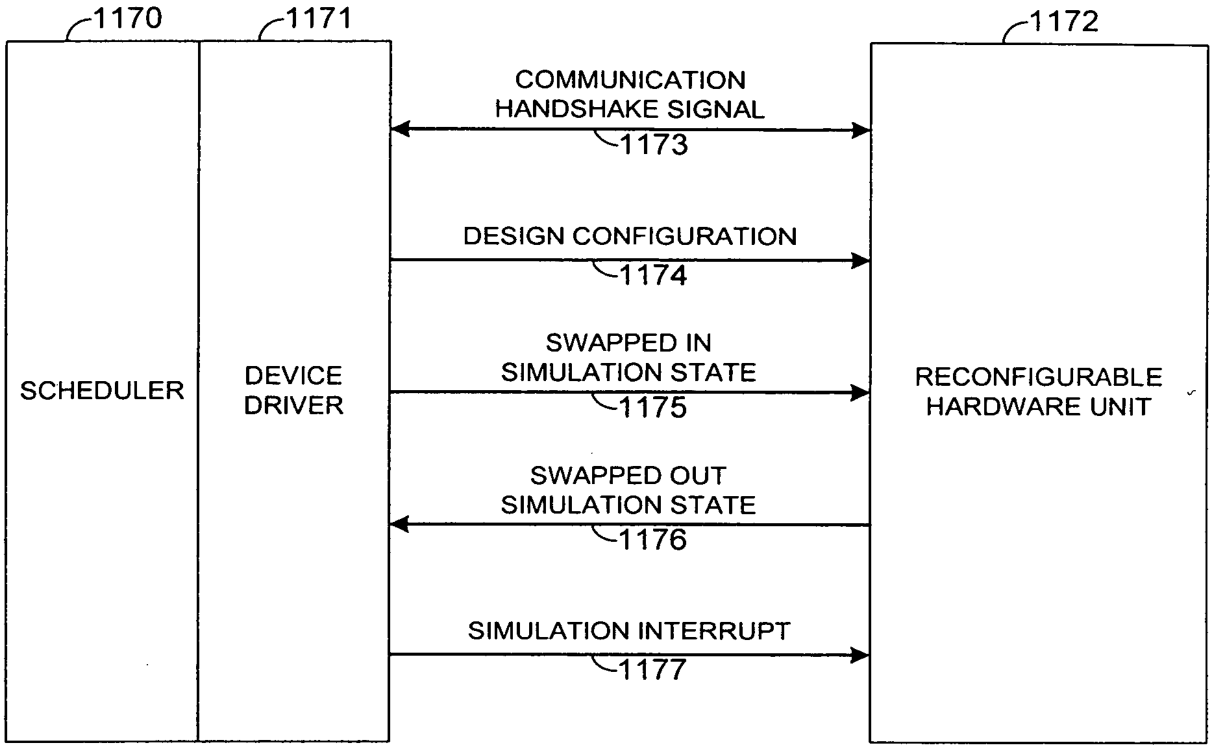

1170
1171
1172
SCHEDULER
DEVICE DRIVER
COMMUNICATION HANDSHAKE SIGNAL
1173
DESIGN CONFIGURATION
1174
SWAPPED IN SIMULATION STATE
1175
SWAPPED OUT SIMULATION STATE
1176
SIMULATION INTERRUPT
1177
RECONFIGURABLE HARDWARE UNIT

FIG. 51

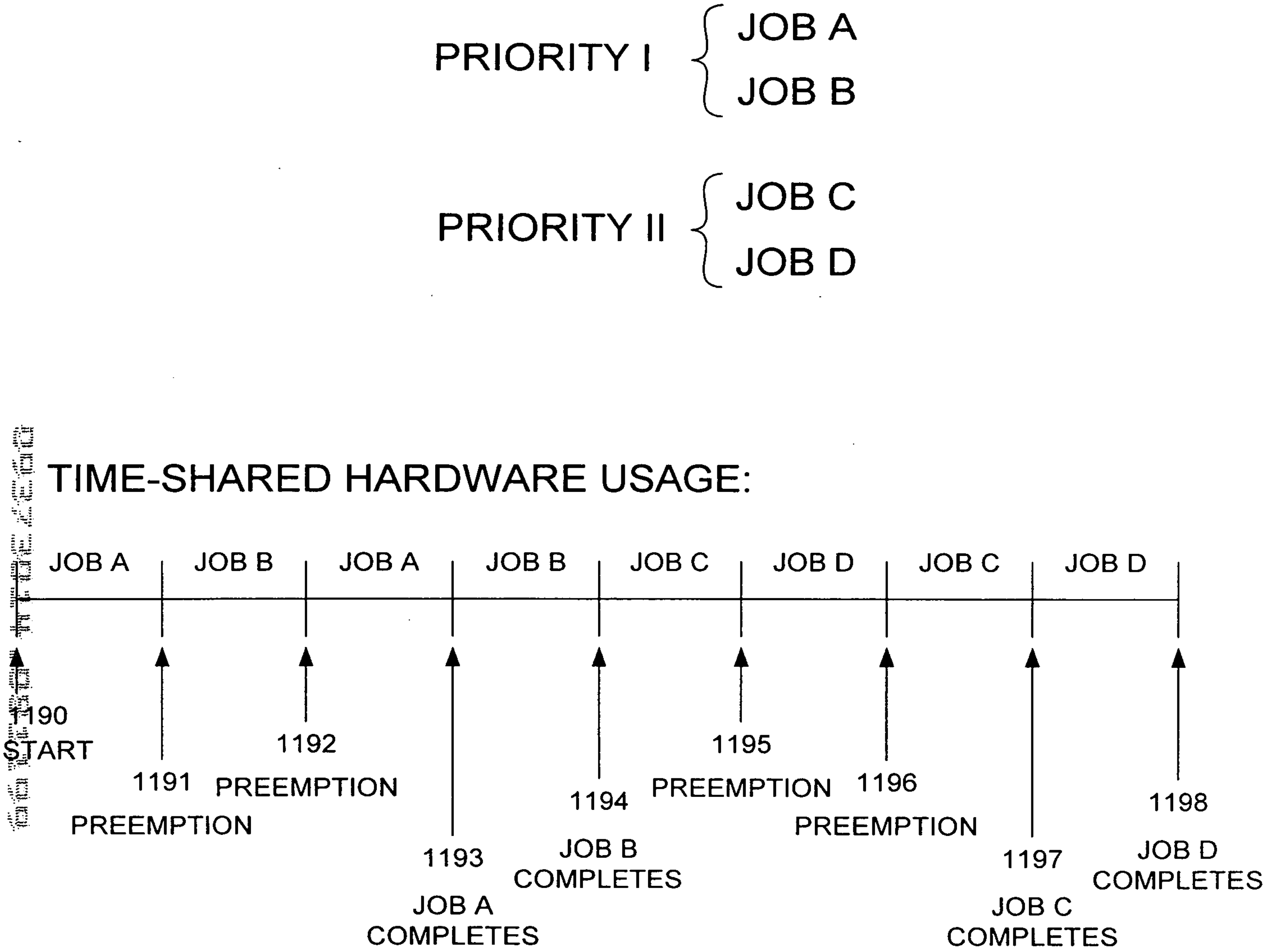

PRIORITY I
JOB A
JOB B
PRIORITY II
JOB C
JOB D
TIME-SHARED HARDWARE USAGE:
JOB A
JOB B
JOB A
JOB B
JOB C
JOB D
JOB C
JOB D
1190
START
1191
PREEMPTION
1192
PREEMPTION
1193
JOB A
COMPLETES
1194
JOB B
COMPLETES
1195
PREEMPTION
1196
PREEMPTION
1197
JOB C
COMPLETES
1198
JOB D
COMPLETES

FIG. 52

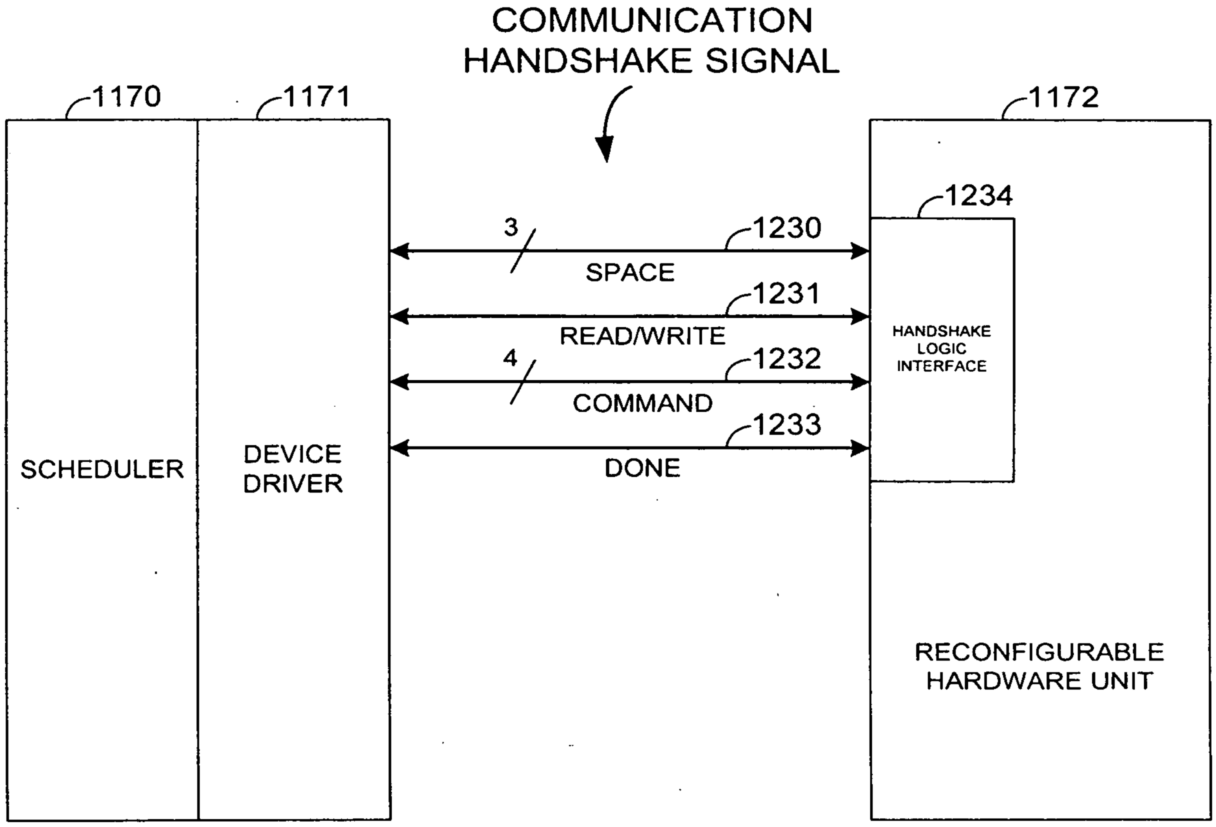

COMMUNICATION
HANDSHAKE SIGNAL
1170
1171
1172
1234
SCHEDULER
DEVICE
DRIVER
3
1230
SPACE
1231
READ/WRITE
4
1232
COMMAND
1233
DONE
HANDSHAKE
LOGIC
INTERFACE
RECONFIGURABLE
HARDWARE UNIT

FIG. 53

COMMUNICATION HANDSHAKE PROTOCOL

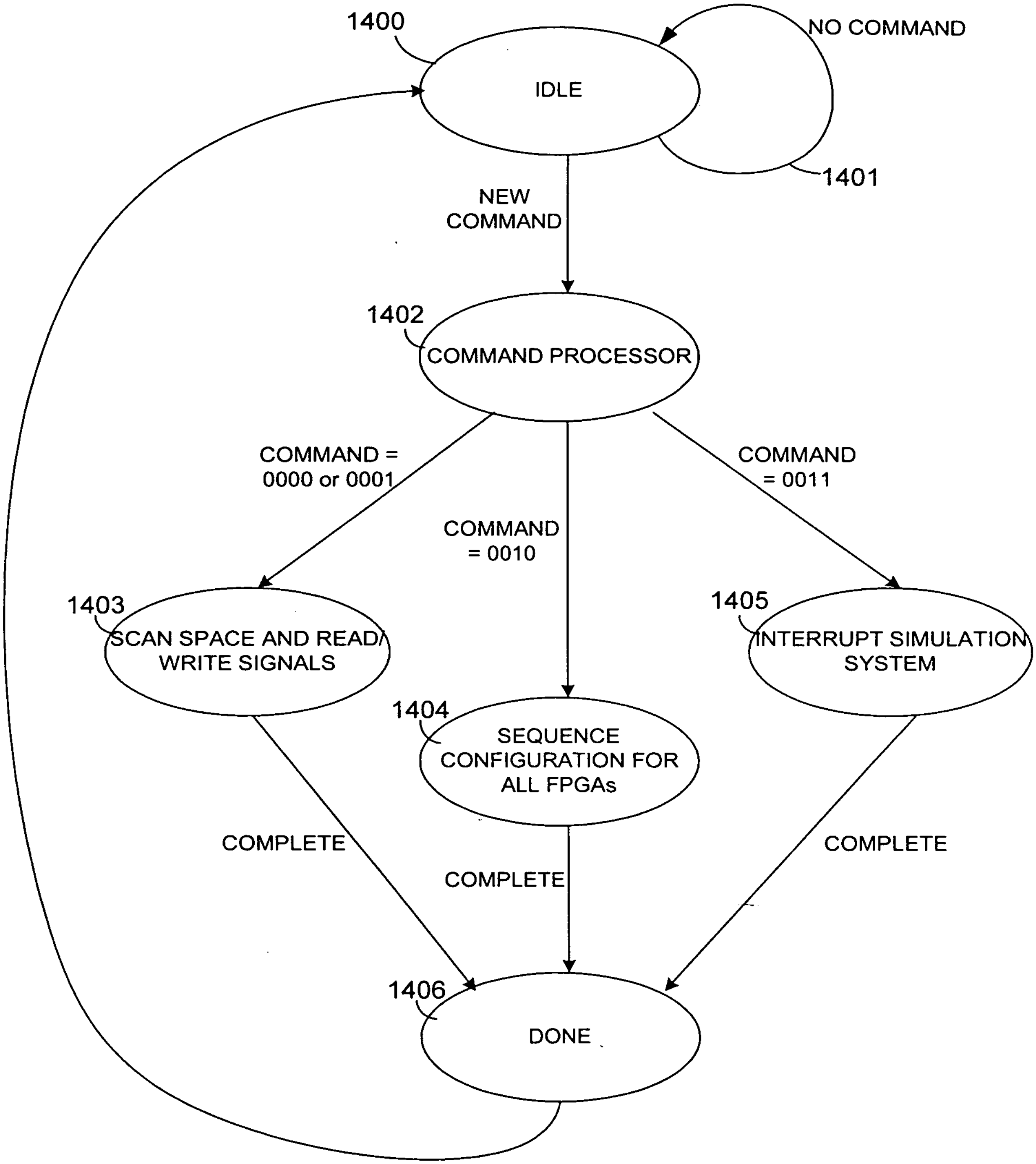


FIG. 54

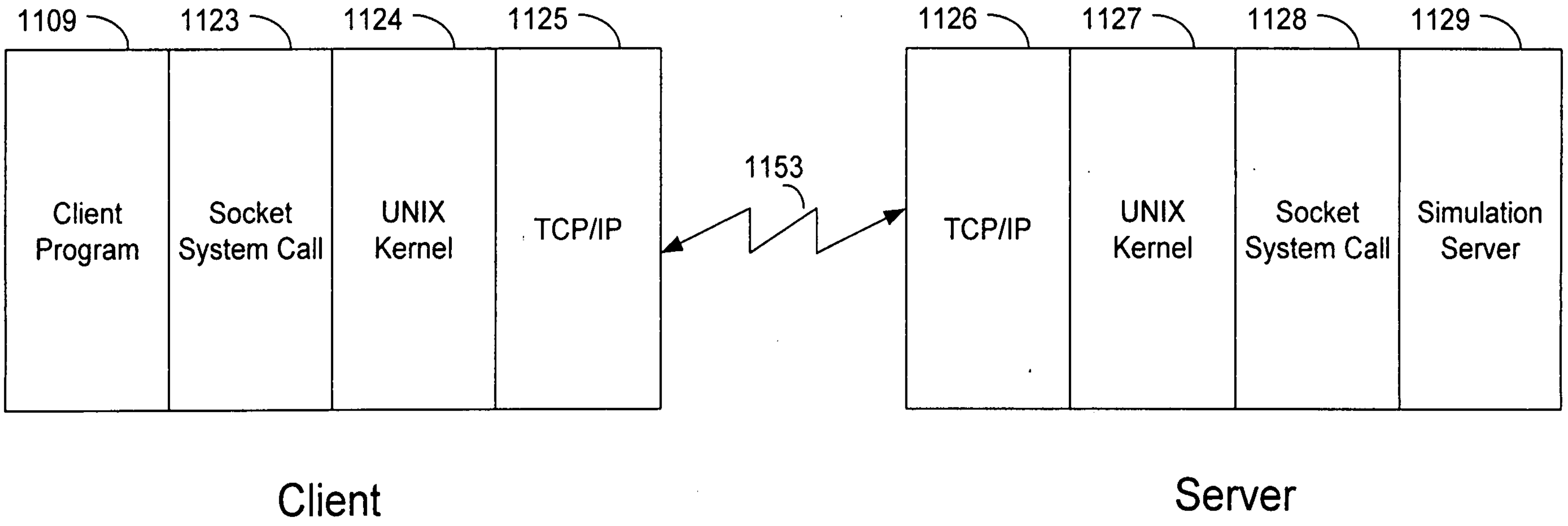
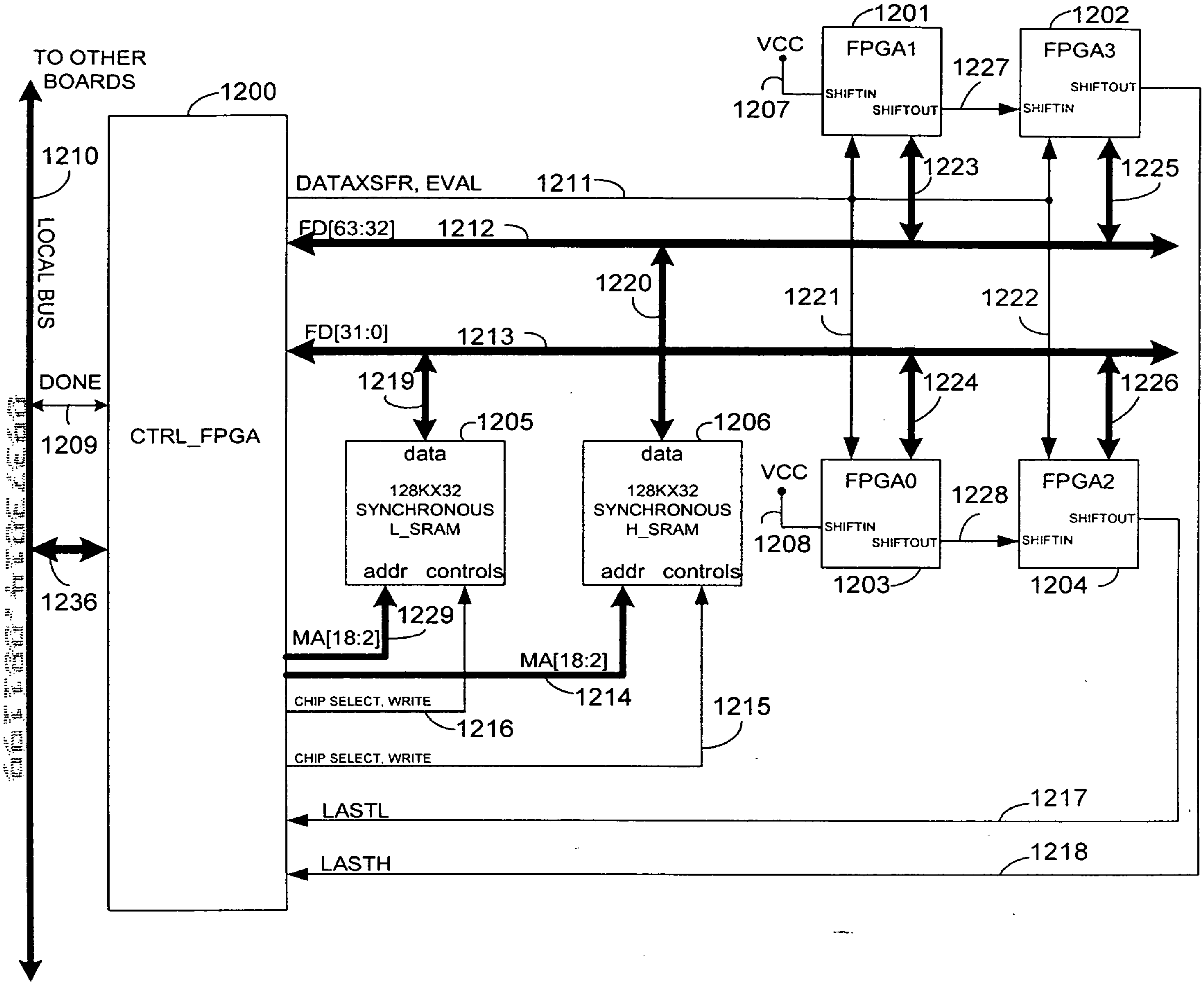

1109
1123
1124
1125
1126
1127
1128
1129
Client Program
Socket System Call
UNIX Kernel
TCP/IP
1153
TCP/IP
UNIX Kernel
Socket System Call
Simulation Server
Client
Server

FIG. 55



TO OTHER BOARDS
1200
1210
LOCAL BUS
DONE
1209
1236
CTRL_FPGA
DATAXSFR, EVAL
1211
FD[63:32]
1212
1220
FD[31:0]
1213
1219
1205
data
128KX32 SYNCHRONOUS L_SRAM
addr
controls
1206
data
128KX32 SYNCHRONOUS H_SRAM
addr
controls
1229
MA[18:2]
MA[18:2]
1214
CHIP SELECT, WRITE
1216
CHIP SELECT, WRITE
1215
LASTL
1217
LASTH
1218
VCC
1207
1201
FPGA1
SHIFTIN
SHIFTOUT
1227
1202
FPGA3
SHIFTOUT
SHIFTIN
1223
1225
1221
1222
1224
1226
VCC
1208
FPGA0
SHIFTIN
SHIFTOUT
1203
1228
FPGA2
SHIFTOUT
SHIFTIN
1204

FIG. 56

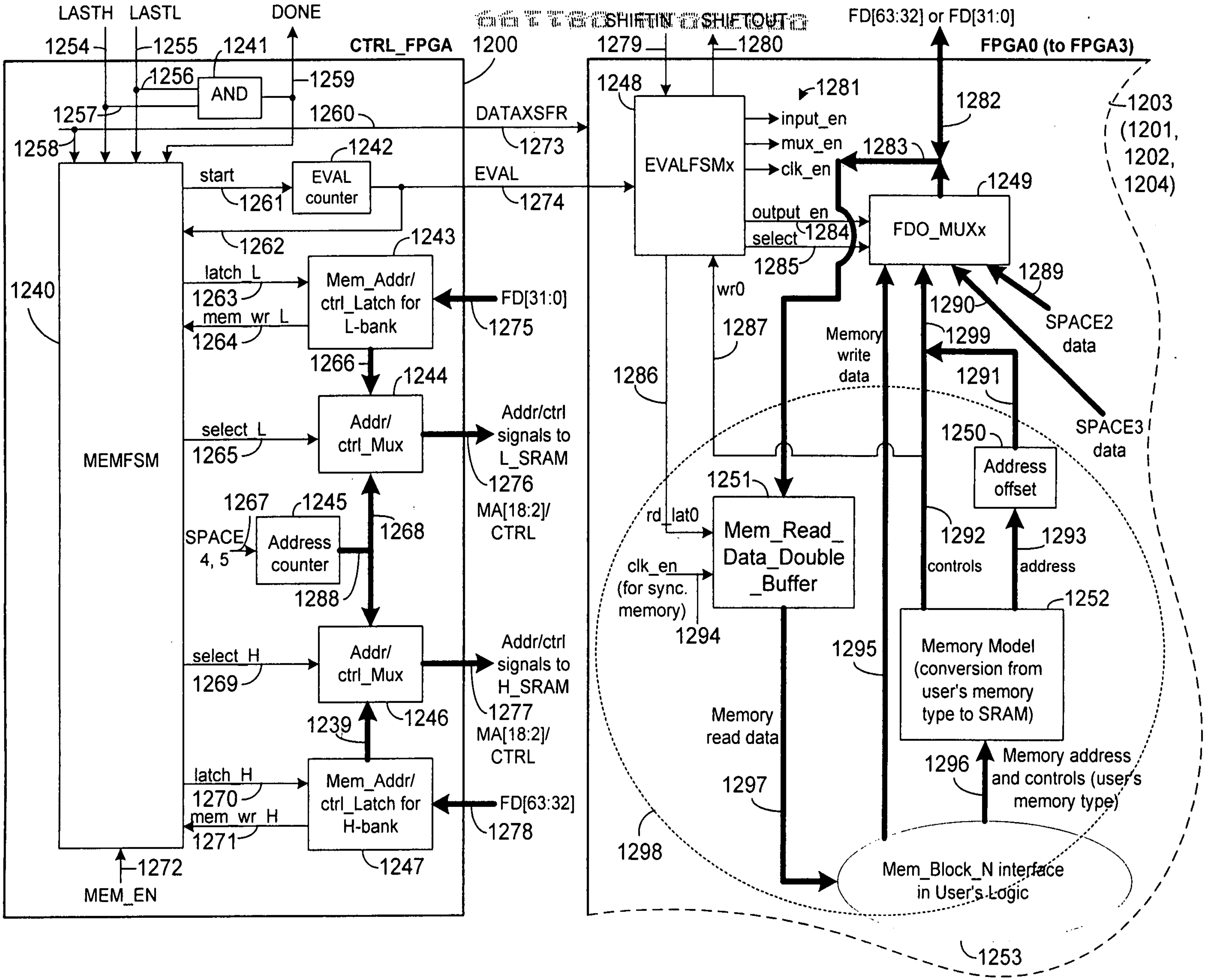
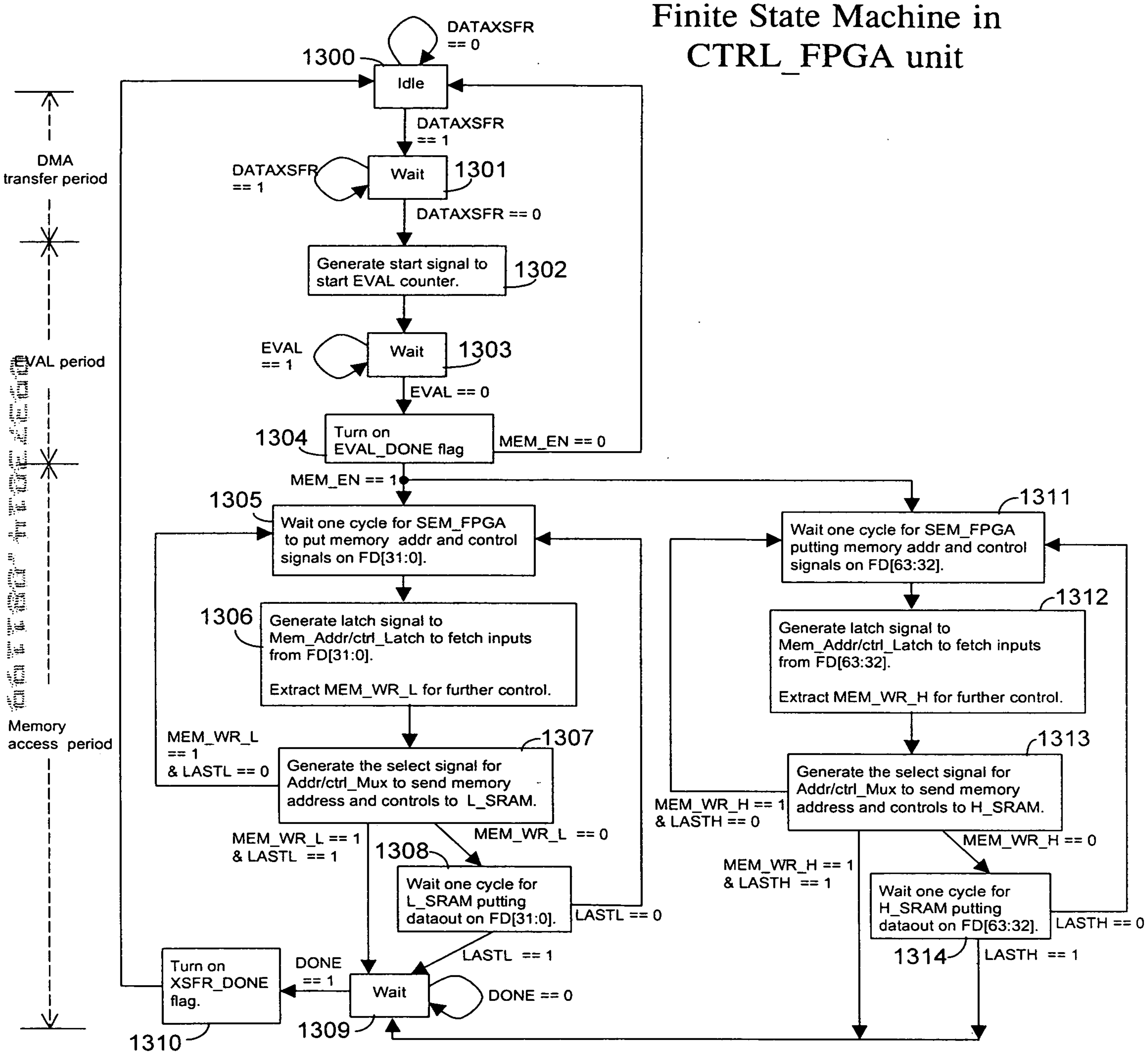

LASTH
LASTL
DONE
1254
1255
1241
CTRL_FPGA
1200
1256
AND
1259
1257
1260
DATAXSFR
1273
1258
1242
start
EVAL counter
EVAL
1274
1261
1262
1243
latch_L
Mem_Addr/ ctrl_Latch for L-bank
FD[31:0]
1263
mem_wr_L
1275
1264
1266
1244
select_L
Addr/ ctrl_Mux
Addr/ctrl signals to L_SRAM
1265
1276
1267
1245
MA[18:2]/ CTRL
SPACE 4, 5
Address counter
1268
1288
select_H
Addr/ ctrl_Mux
Addr/ctrl signals to H_SRAM
1269
1277
1239
1246
MA[18:2]/ CTRL
latch_H
Mem_Addr/ ctrl_Latch for H-bank
FD[63:32]
1270
mem_wr_H
1278
1271
1247
1272
MEM_EN
1240
MEMFSM
SHIFTIN
SHIFTOUT
FD[63:32] or FD[31:0]
1279
1280
FPGA0 (to FPGA3)
1248
1281
1282
1203
(1201, 1202, 1204)
input_en
mux_en
clk_en
1283
EVALFSMx
1249
output_en
1284
select
FDO_MUXx
1285
1289
wr0
1290
SPACE2 data
1287
1299
1286
Memory write data
1291
SPACE3 data
1250
Address offset
1251
rd_lat0
Mem_Read_ Data_Double _Buffer
1292
1293
clk_en (for sync. memory)
controls
address
1294
1252
1295
Memory Model (conversion from user's memory type to SRAM)
Memory read data
1296
Memory address and controls (user's memory type)
1297
1298
Mem_Block_N interface in User's Logic
1253

FIG. 57

MEMFSM - Memory Finite State Machine in CTRL_FPGA unit



DATAXSFR == 0
1300
Idle
DMA transfer period
DATAXSFR == 1
DATAXSFR == 1
Wait
1301
DATAXSFR == 0
Generate start signal to start EVAL counter.
1302
EVAL == 1
Wait
1303
EVAL period
EVAL == 0
1304
Turn on EVAL_DONE flag
MEM_EN == 0
MEM_EN == 1
1305
Wait one cycle for SEM_FPGA to put memory addr and control signals on FD[31:0].
1306
Generate latch signal to Mem_Addr/ctrl_Latch to fetch inputs from FD[31:0].
Extract MEM_WR_L for further control.
Memory access period
MEM_WR_L == 1 & LASTL == 0
1307
Generate the select signal for Addr/ctrl_Mux to send memory address and controls to L_SRAM.
MEM_WR_L == 1 & LASTL == 1
MEM_WR_L == 0
1308
Wait one cycle for L_SRAM putting dataout on FD[31:0].
LASTL == 0
LASTL == 1
Turn on XSFR_DONE flag.
DONE == 1
Wait
DONE == 0
1310
1309
1311
Wait one cycle for SEM_FPGA putting memory addr and control signals on FD[63:32].
1312
Generate latch signal to Mem_Addr/ctrl_Latch to fetch inputs from FD[63:32].
Extract MEM_WR_H for further control.
1313
Generate the select signal for Addr/ctrl_Mux to send memory address and controls to H_SRAM.
MEM_WR_H == 1 & LASTH == 0
MEM_WR_H == 0
MEM_WR_H == 1 & LASTH == 1
Wait one cycle for H_SRAM putting dataout on FD[63:32].
LASTH == 0
1314
LASTH == 1

FIG. 58

EVALFSM - EVAL
Finite State Machine in
each FPGA logic device

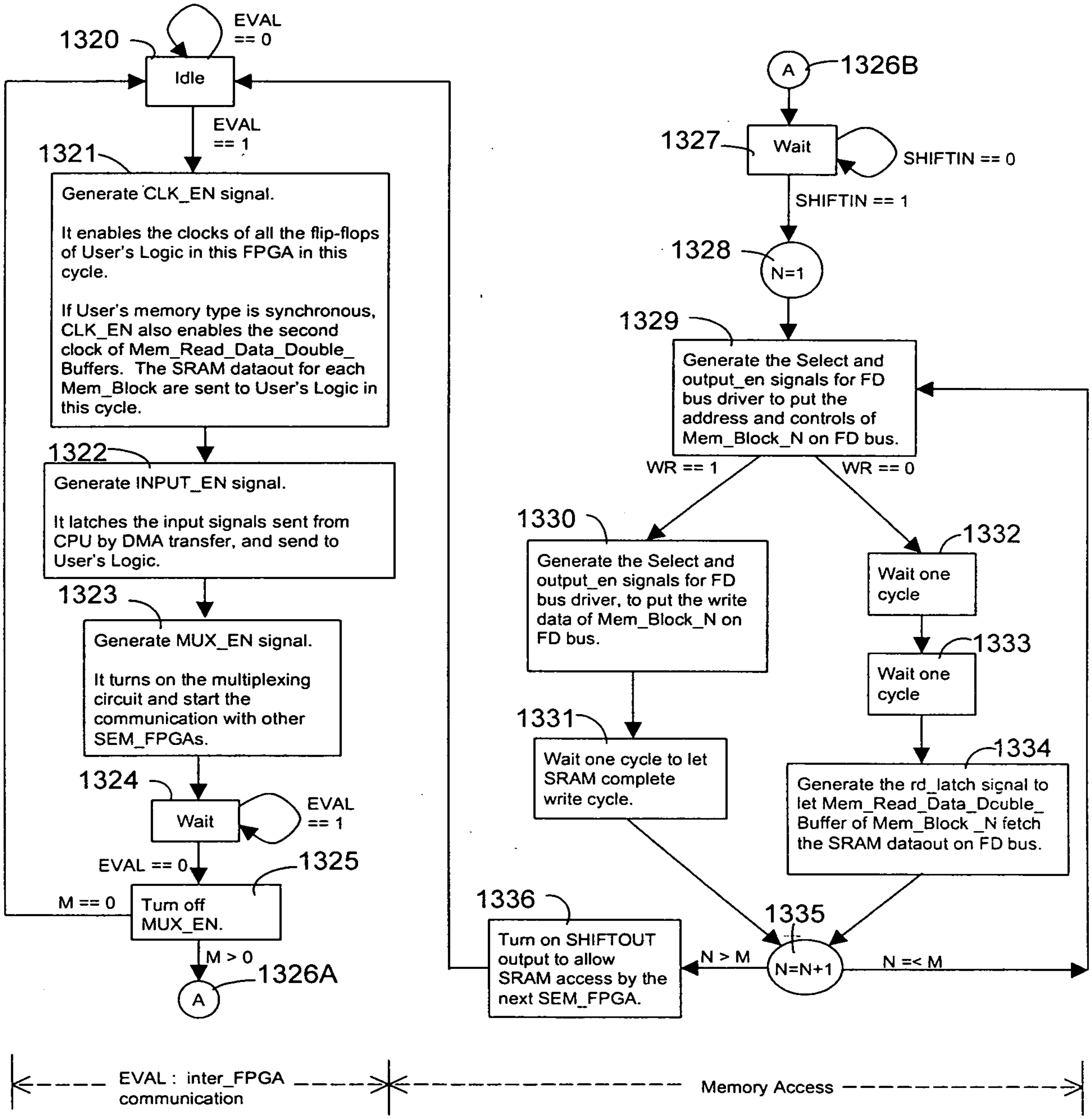

1320
EVAL == 0
Idle
EVAL == 1
1321
Generate CLK_EN signal.
It enables the clocks of all the flip-flops of User's Logic in this FPGA in this cycle.
If User's memory type is synchronous, CLK_EN also enables the second clock of Mem_Read_Data_Double_ Buffers. The SRAM dataout for each Mem_Block are sent to User's Logic in this cycle.
1322
Generate INPUT_EN signal.
It latches the input signals sent from CPU by DMA transfer, and send to User's Logic.
1323
Generate MUX_EN signal.
It turns on the multiplexing circuit and start the communication with other SEM_FPGAs.
1324
Wait
EVAL == 1
EVAL == 0
1325
M == 0
Turn off MUX_EN.
M > 0
1326A
A
A
1326B
1327
Wait
SHIFTIN == 0
SHIFTIN == 1
1328
N=1
1329
Generate the Select and output_en signals for FD bus driver to put the address and controls of Mem_Block_N on FD bus.
WR == 1
WR == 0
1330
Generate the Select and output_en signals for FD bus driver, to put the write data of Mem_Block_N on FD bus.
1332
Wait one cycle
1333
Wait one cycle
1331
Wait one cycle to let SRAM complete write cycle.
1334
Generate the rd_latch signal to let Mem_Read_Data_Dcuble_ Buffer of Mem_Block _N fetch the SRAM dataout on FD bus.
1336
Turn on SHIFTOUT output to allow SRAM access by the next SEM_FPGA.
N > M
1335
N=N+1
N =< M
EVAL : inter_FPGA communication
Memory Access

FIG. 59

MEMORY READ DATA DOUBLE BUFFER

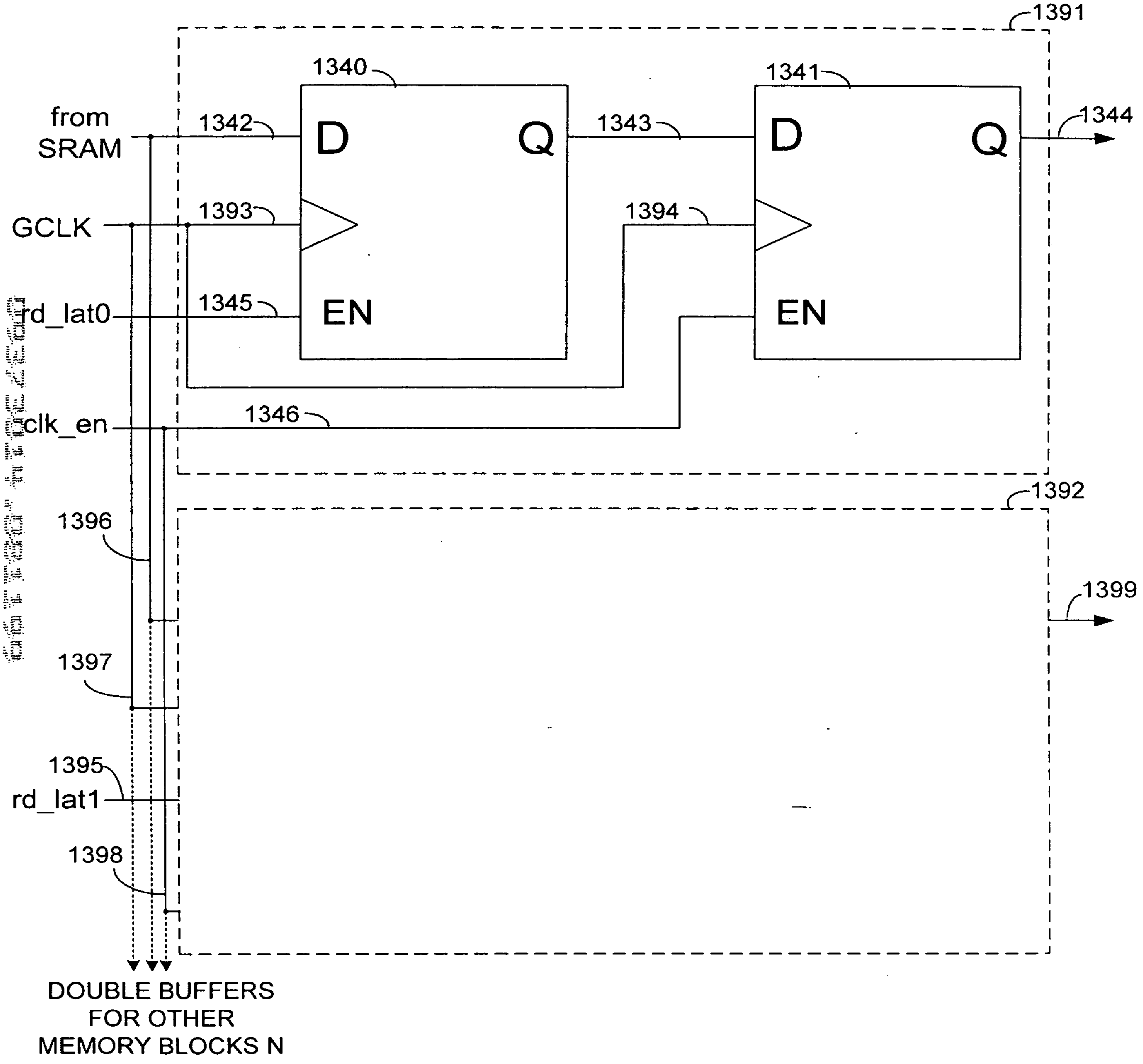


FIG. 60

SIMULATION WRITE/READ CYCLE

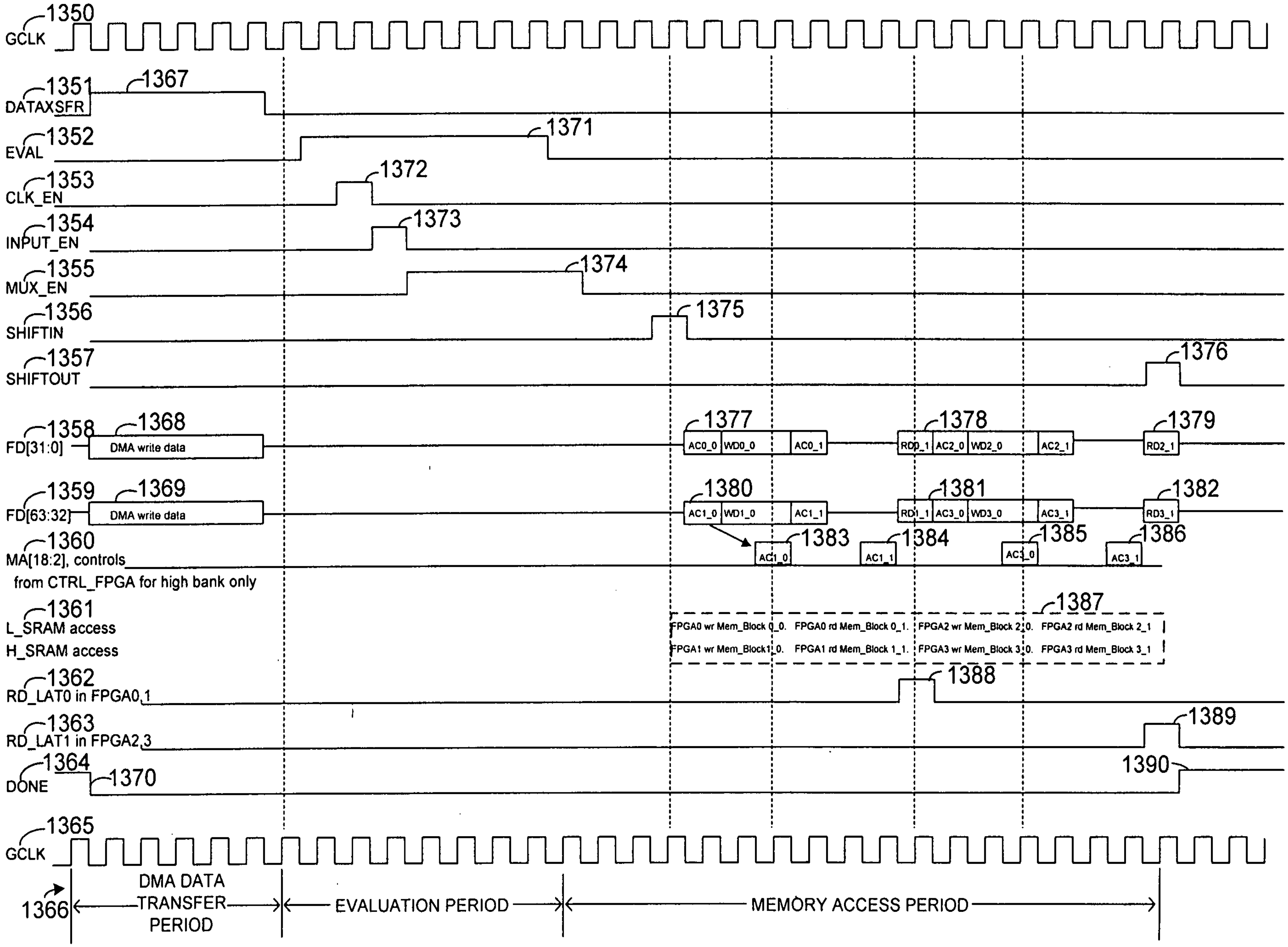


FIG. 61

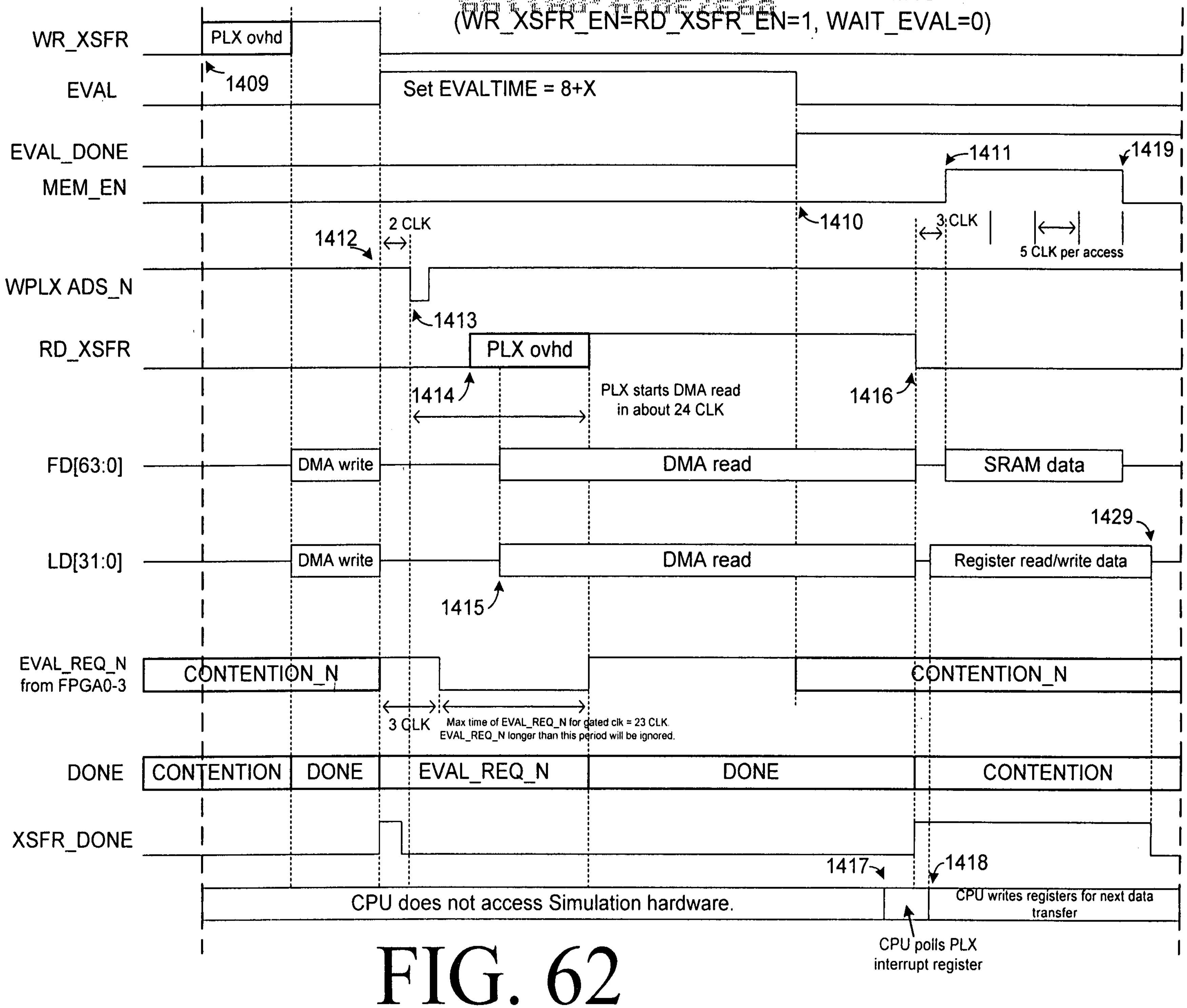

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)
WR_XSFR
PLX ovhd
1409
EVAL
Set EVALTIME = 8+X
EVAL_DONE
MEM_EN
1411
1419
1410
1412
2 CLK
3 CLK
5 CLK per access
WPLX ADS_N
1413
RD_XSFR
PLX ovhd
1414
PLX starts DMA read
in about 24 CLK
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1415
EVAL_REQ_N
from FPGA0-3
CONTENTION_N
CONTENTION_N
3 CLK
Max time of EVAL_REQ_N for gated clk = 23 CLK.
EVAL_REQ_N longer than this period will be ignored.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1417
1418
CPU does not access Simulation hardware.
CPU writes registers for next data transfer
CPU polls PLX
interrupt register

FIG. 62

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=1)

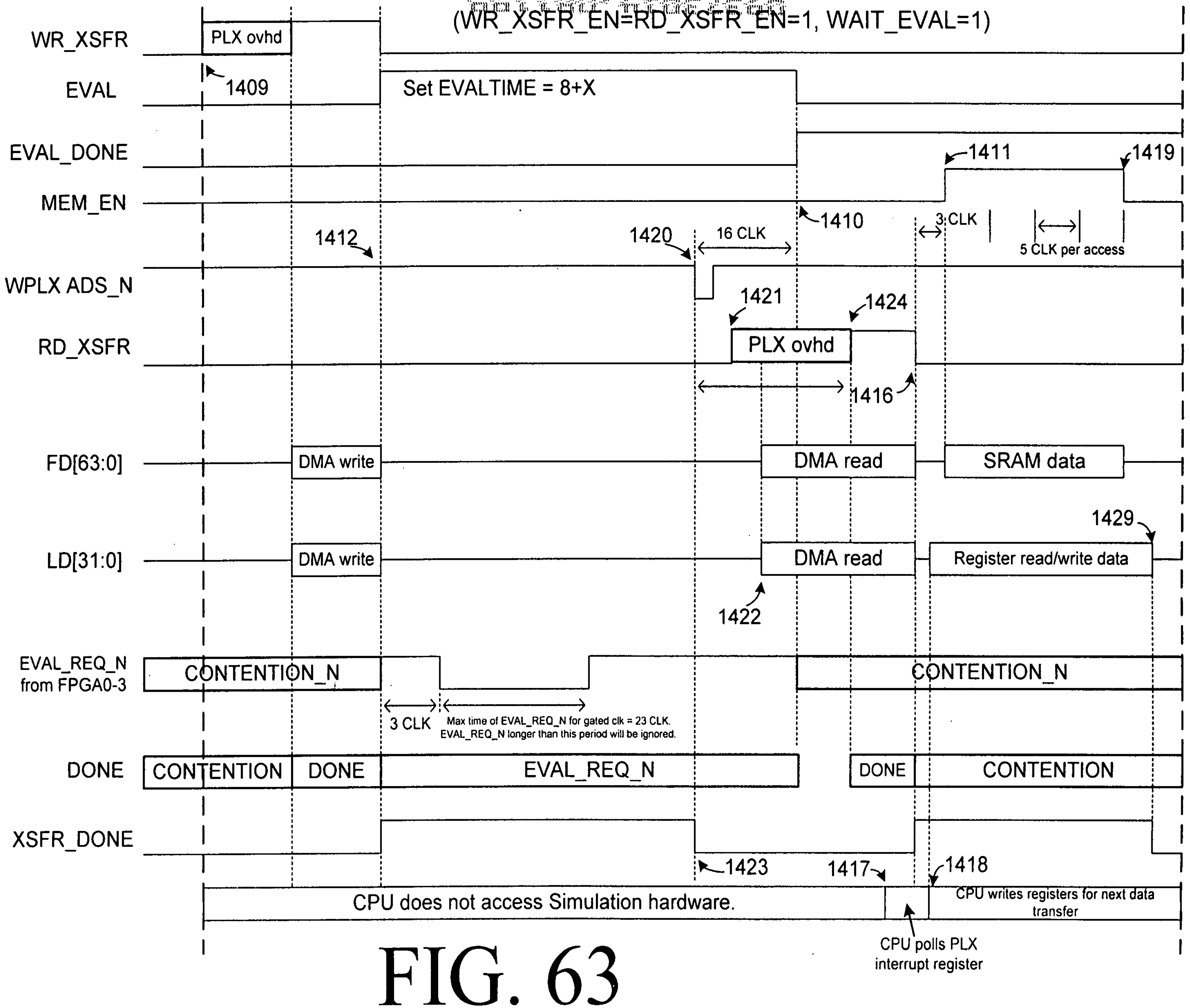
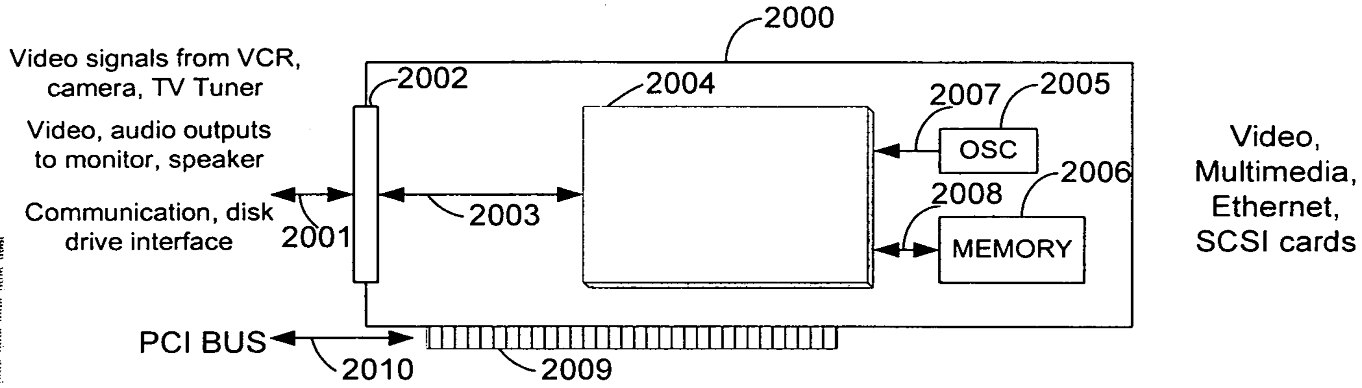


FIG. 63

Typical User Design of PCI Add-on Cards



2000
Video signals from VCR, camera, TV Tuner
Video, audio outputs to monitor, speaker
Communication, disk drive interface
2001
2002
2003
2004
2007
2005
OSC
2006
2008
MEMORY
Video, Multimedia, Ethernet, SCSI cards
PCI BUS
2010
2009

FIG. 64

Typical Hardware/Software Co-Verification

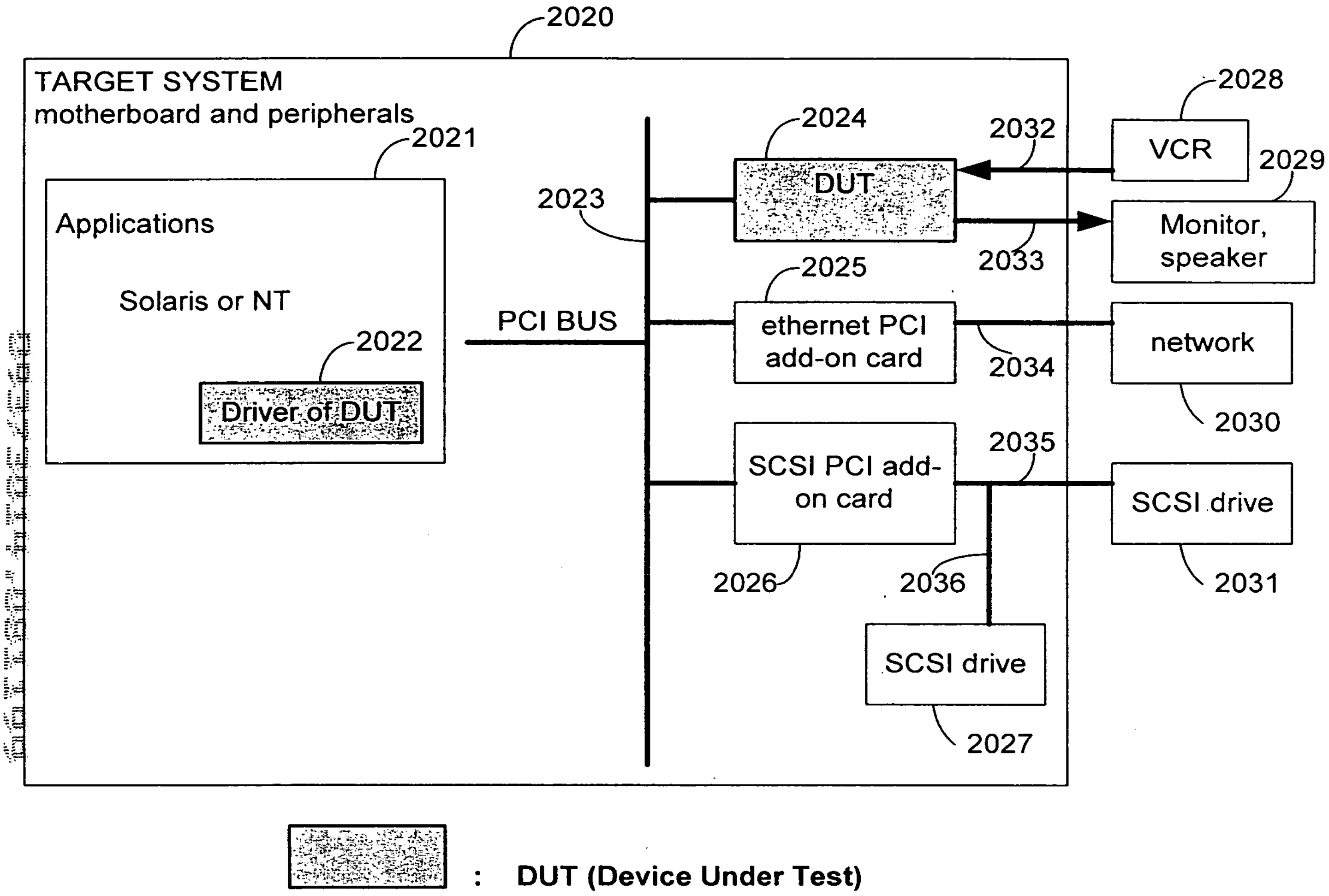
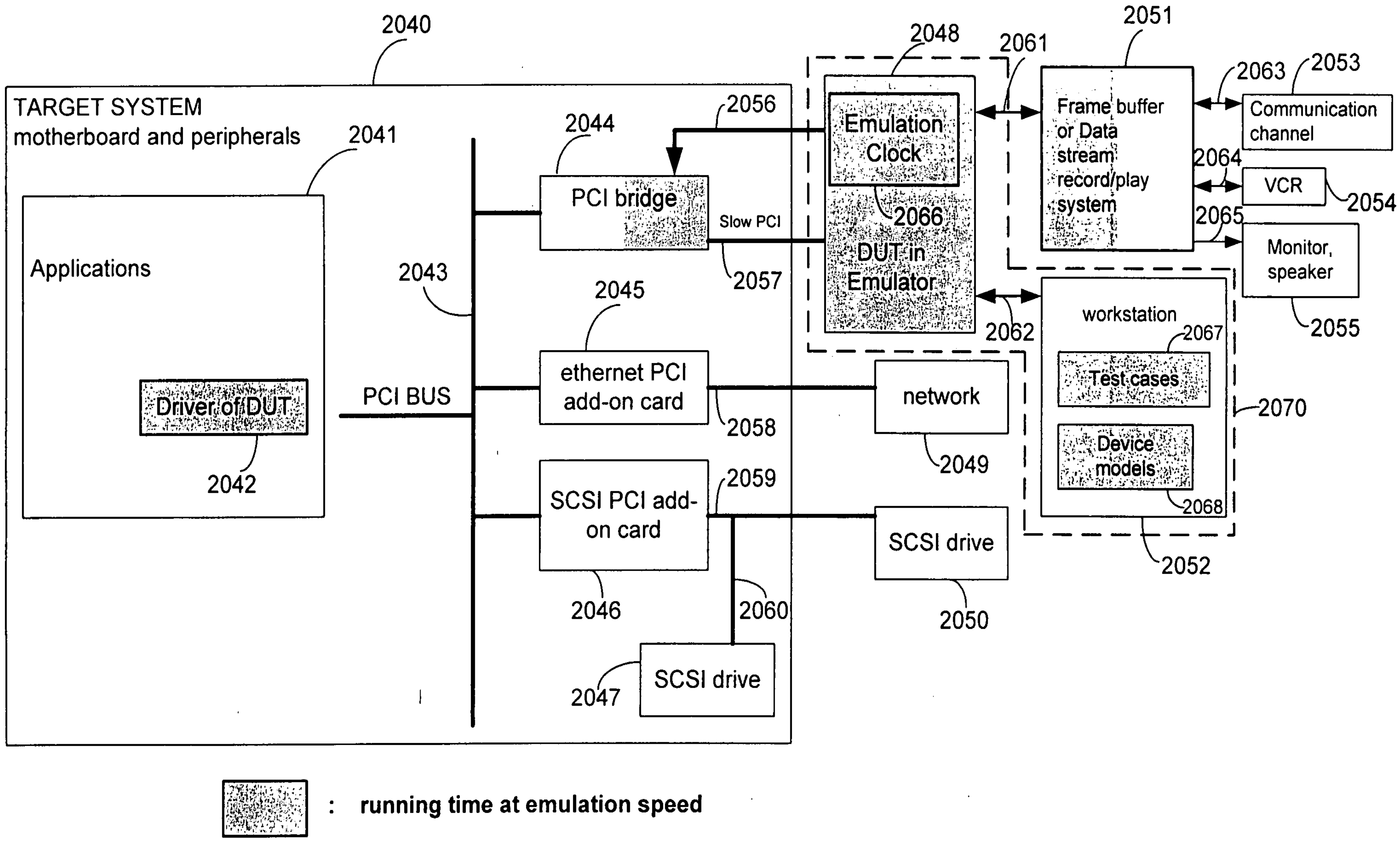


FIG. 65

Typical Co-Verification by Using Emulator



The rest of the target system is running at full speed.

FIG. 66

SIMULATION

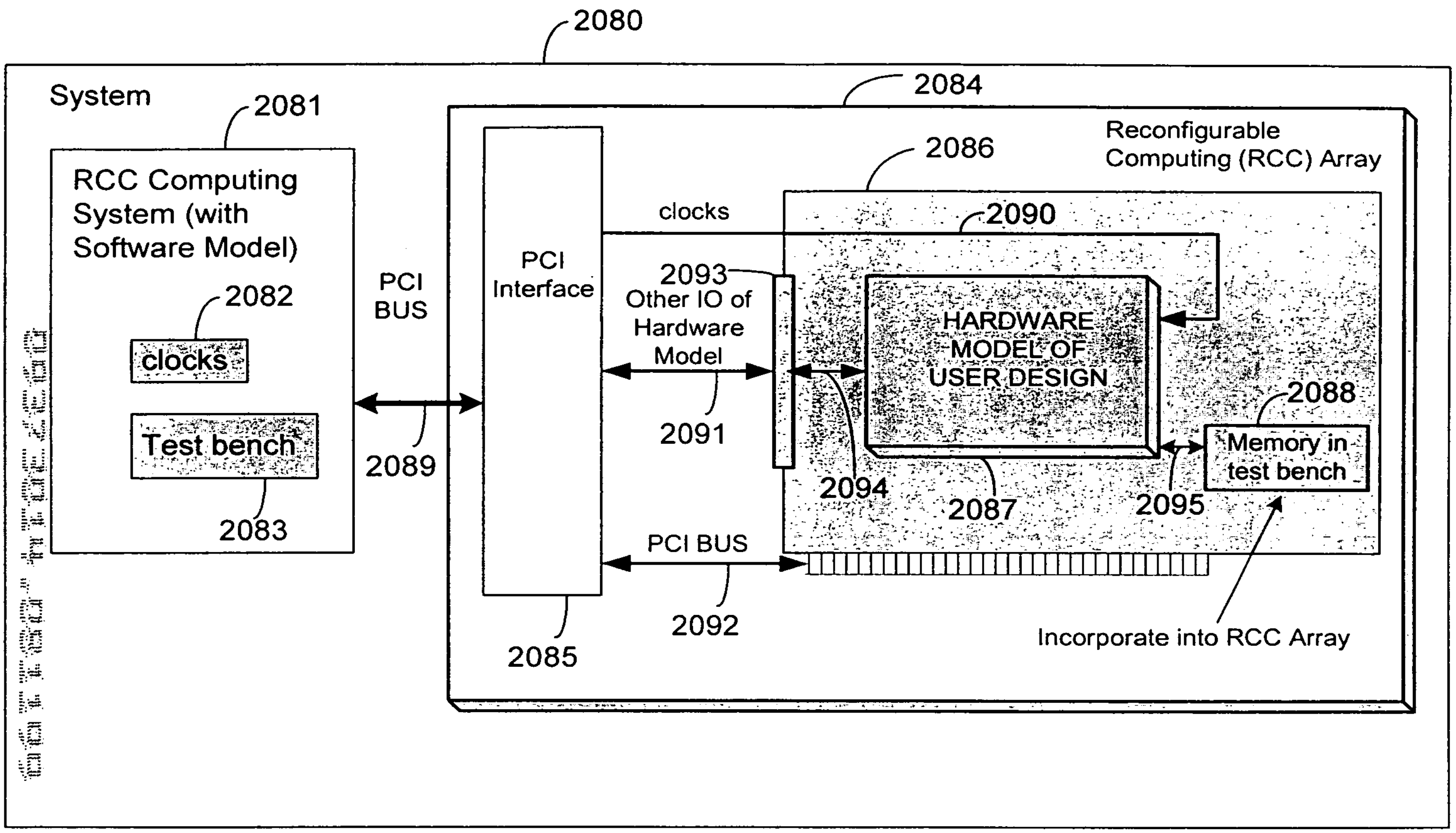

2080
System
2081
RCC Computing System (with Software Model)
2082
clocks
Test bench
2083
PCI BUS
2089
2084
Reconfigurable Computing (RCC) Array
PCI Interface
2085
clocks
2086
2090
2093
Other IO of Hardware Model
2091
HARDWARE MODEL OF USER DESIGN
2094
2087
2088
Memory in test bench
2095
PCI BUS
2092
Incorporate into RCC Array

FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

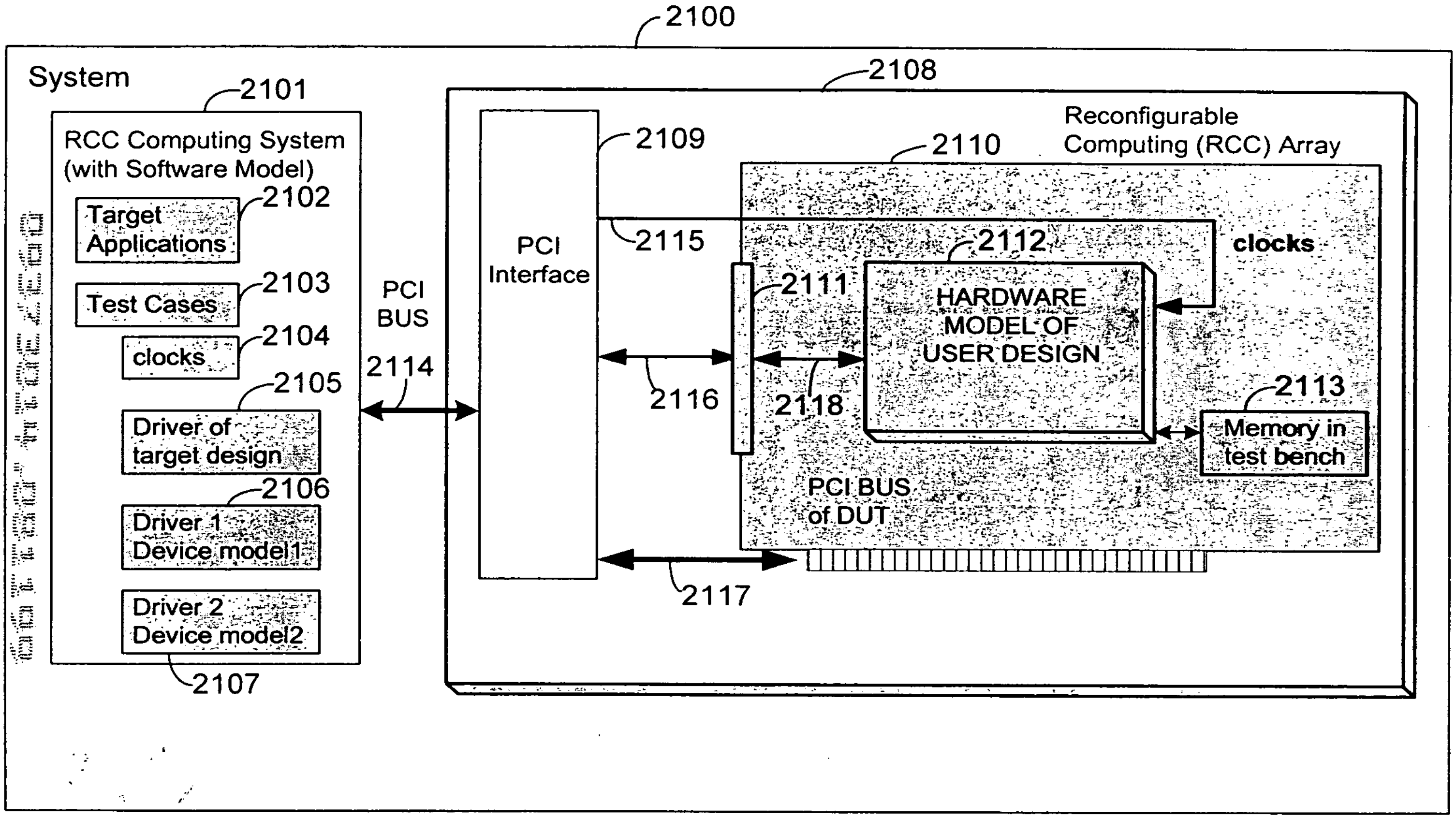


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

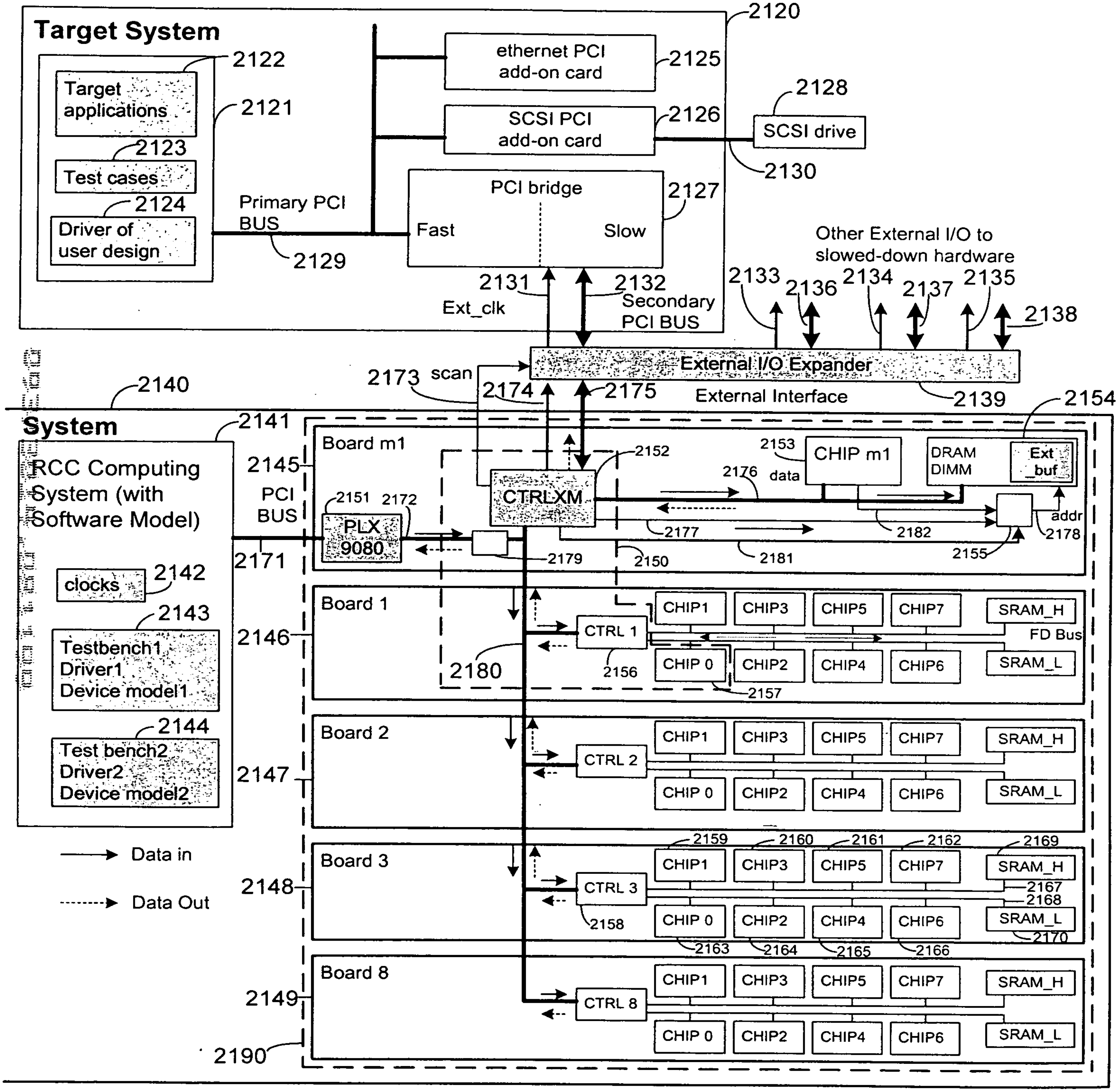


FIG. 69

CONTROL OF DATA-IN CYCLE

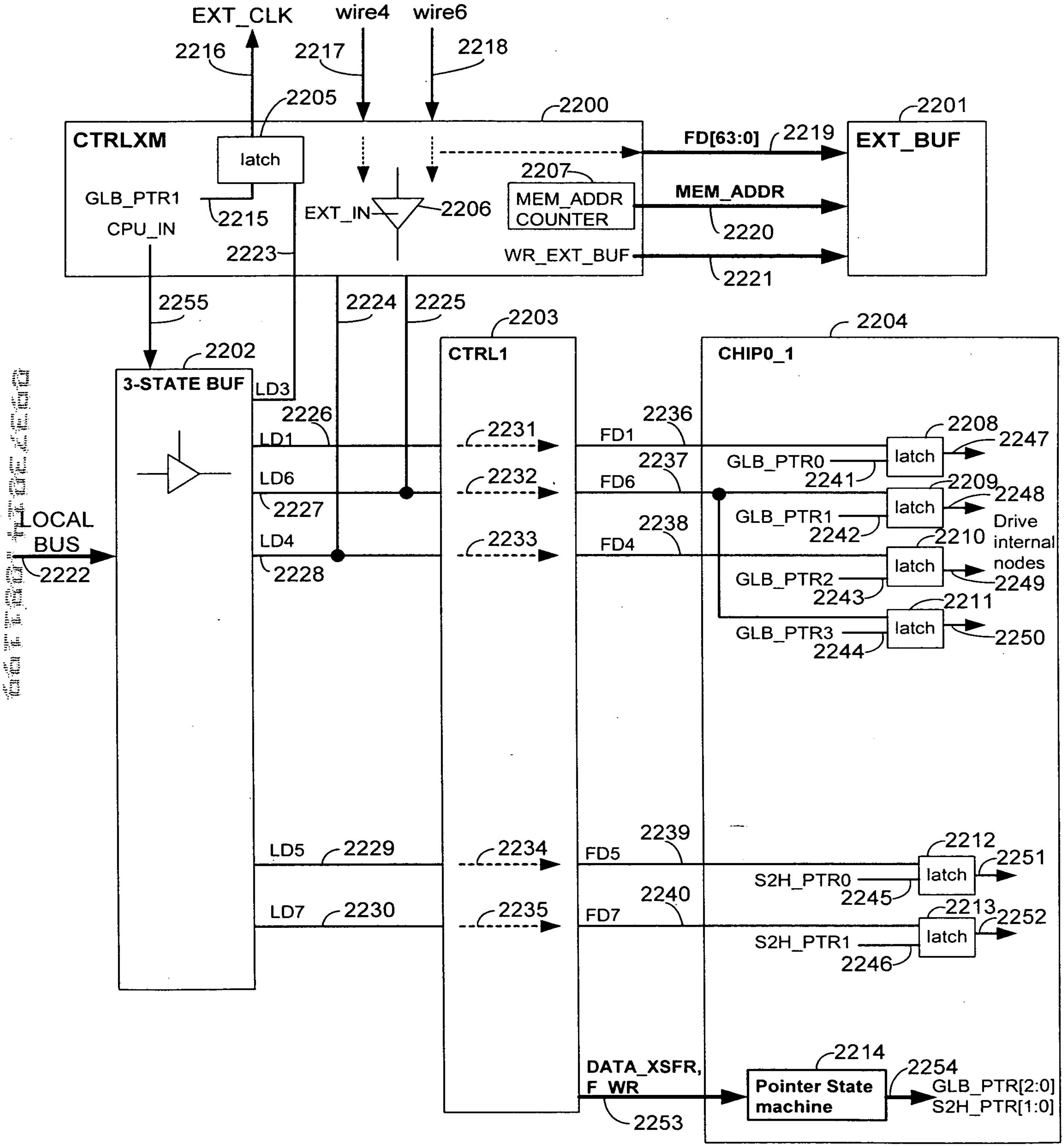


FIG. 70

CONTROL OF DATA-OUT CYCLE

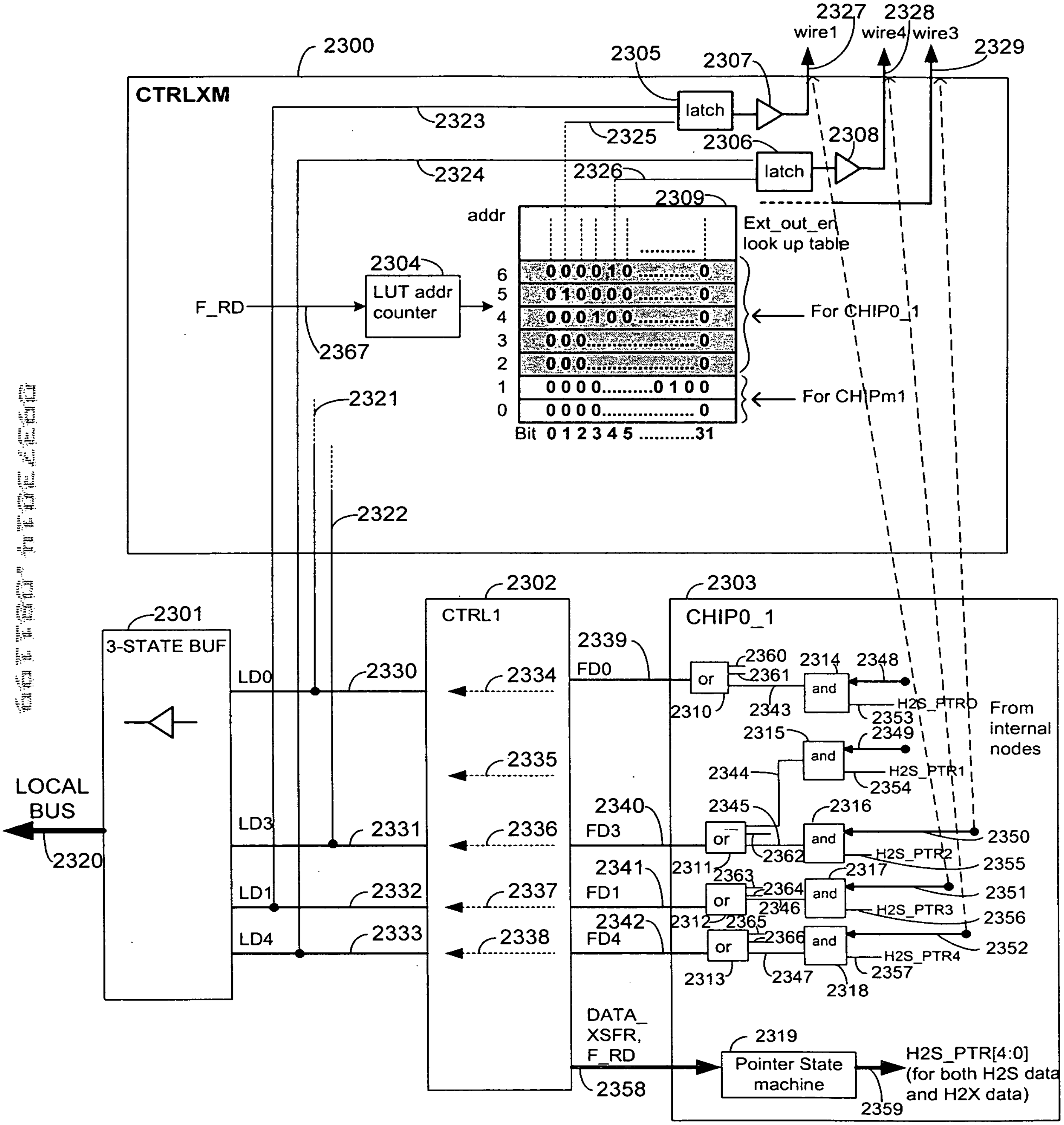


FIG. 71

CONTROL OF DATA-IN CYCLE

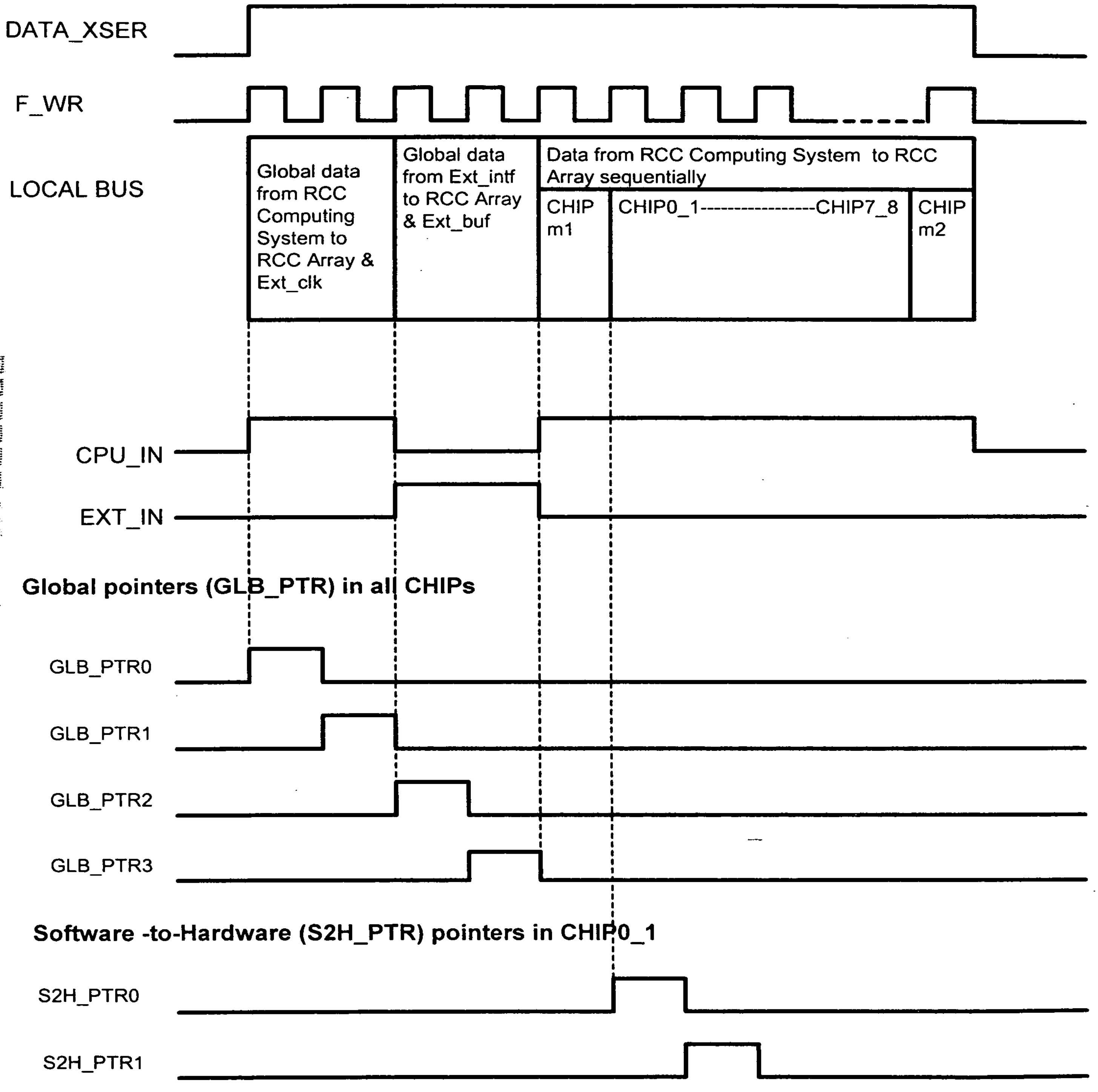

DATA_XSER
F_WR
LOCAL BUS
Global data from RCC Computing System to RCC Array & Ext_clk
Global data from Ext_intf to RCC Array & Ext_buf
Data from RCC Computing System to RCC Array sequentially
CHIP m1
CHIP0_1-----------------CHIP7_8
CHIP m2
CPU_IN
EXT_IN
Global pointers (GLB_PTR) in all CHIPs
GLB_PTR0
GLB_PTR1
GLB_PTR2
GLB_PTR3
Software -to-Hardware (S2H_PTR) pointers in CHIP0_1
S2H_PTR0
S2H_PTR1

FIG. 72

CONTROL OF DATA-OUT CYCLE

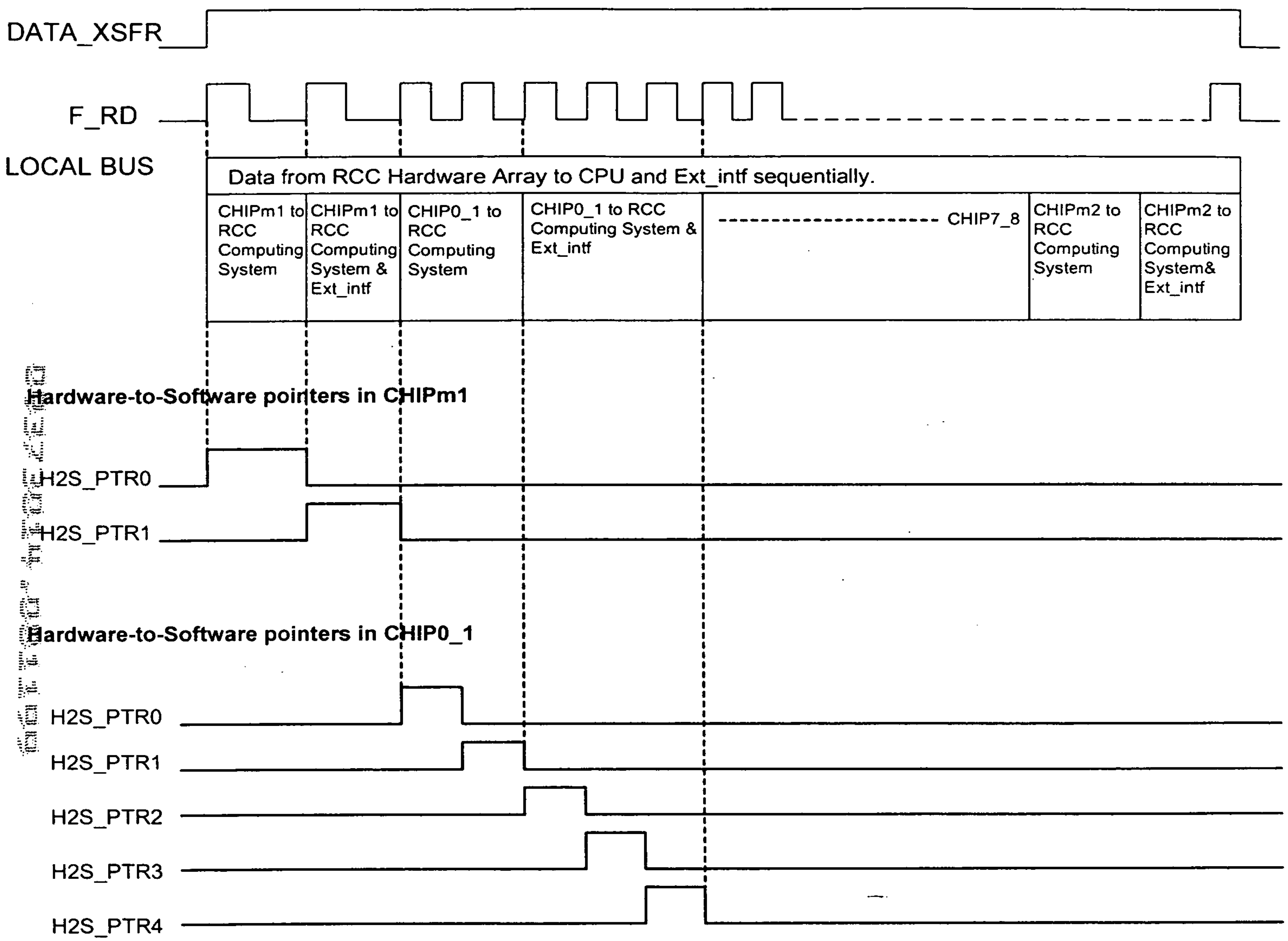

DATA_XSFR
F_RD
LOCAL BUS
Data from RCC Hardware Array to CPU and Ext_intf sequentially.
CHIPm1 to RCC Computing System
CHIPm1 to RCC Computing System & Ext_intf
CHIP0_1 to RCC Computing System
CHIP0_1 to RCC Computing System & Ext_intf
CHIP7_8
CHIPm2 to RCC Computing System
CHIPm2 to RCC Computing System& Ext_intf
Hardware-to-Software pointers in CHIPm1
H2S_PTR0
H2S_PTR1
Hardware-to-Software pointers in CHIP0_1
H2S_PTR0
H2S_PTR1
H2S_PTR2
H2S_PTR3
H2S_PTR4

FIG. 73

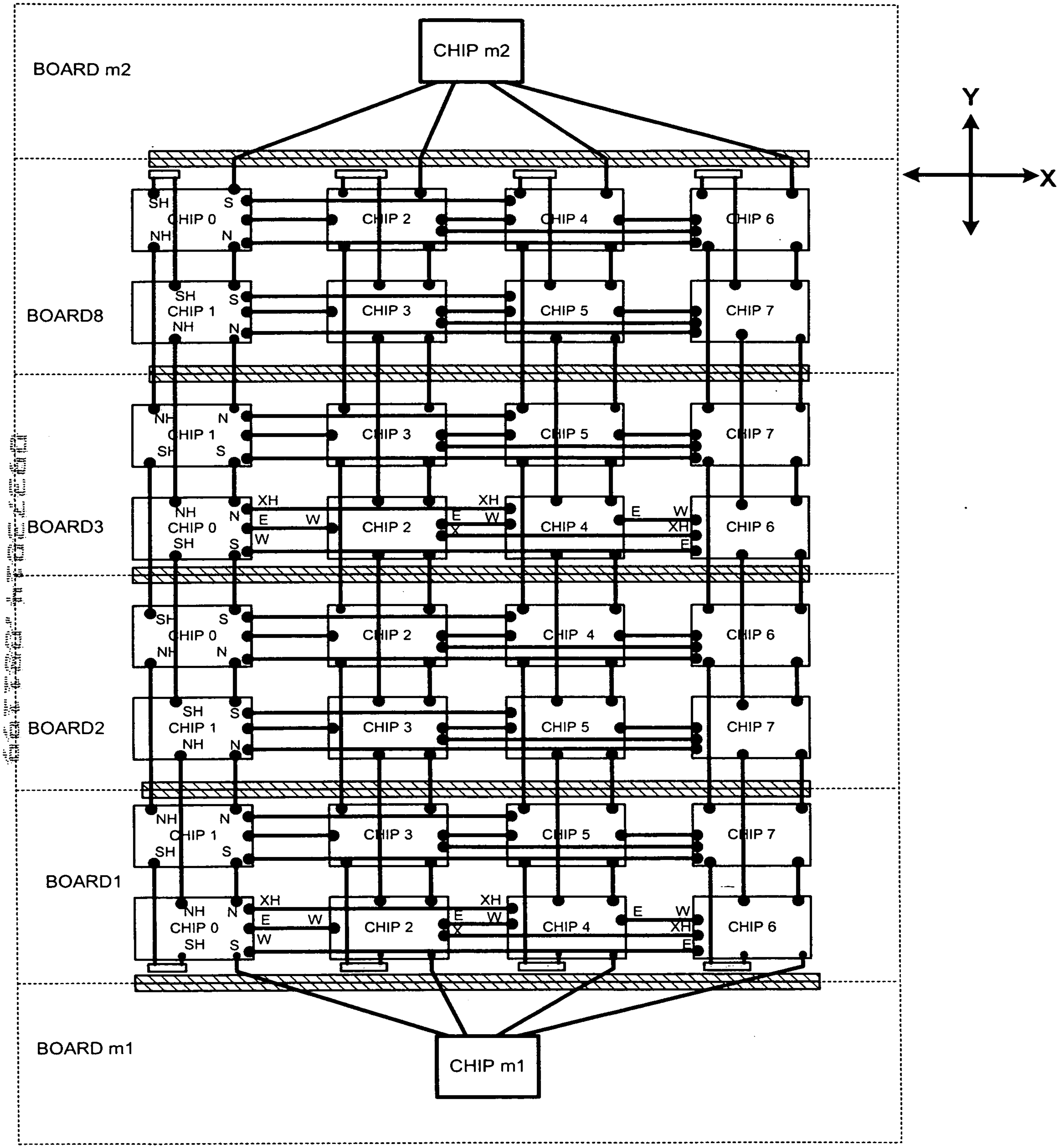

BOARD m2
CHIP m2
Y
X
BOARD8
BOARD3
BOARD2
BOARD1
BOARD m1
CHIP m1
CHIP 0
CHIP 1
CHIP 2
CHIP 3
CHIP 4
CHIP 5
CHIP 6
CHIP 7
SH
NH
S
N
XH
E
W
X

FIG. 74

SHIFT REGISTER

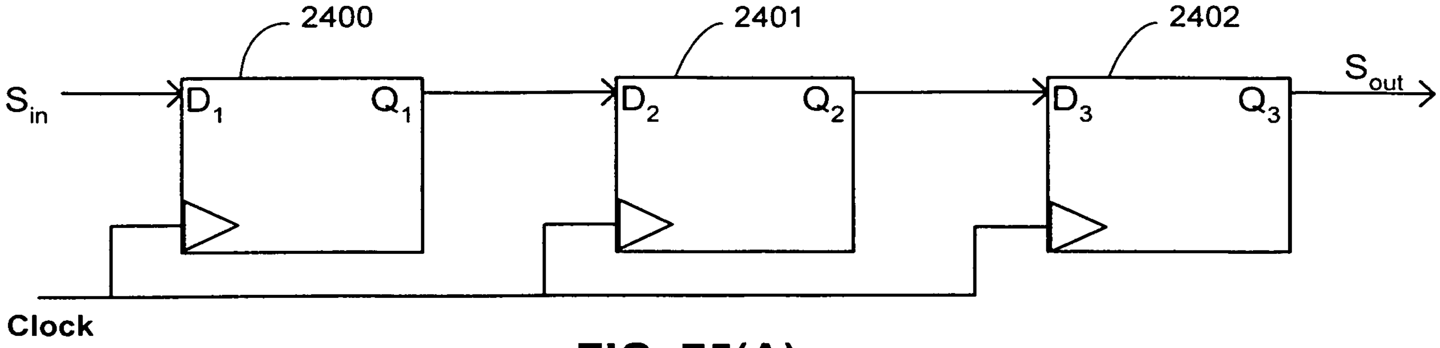


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

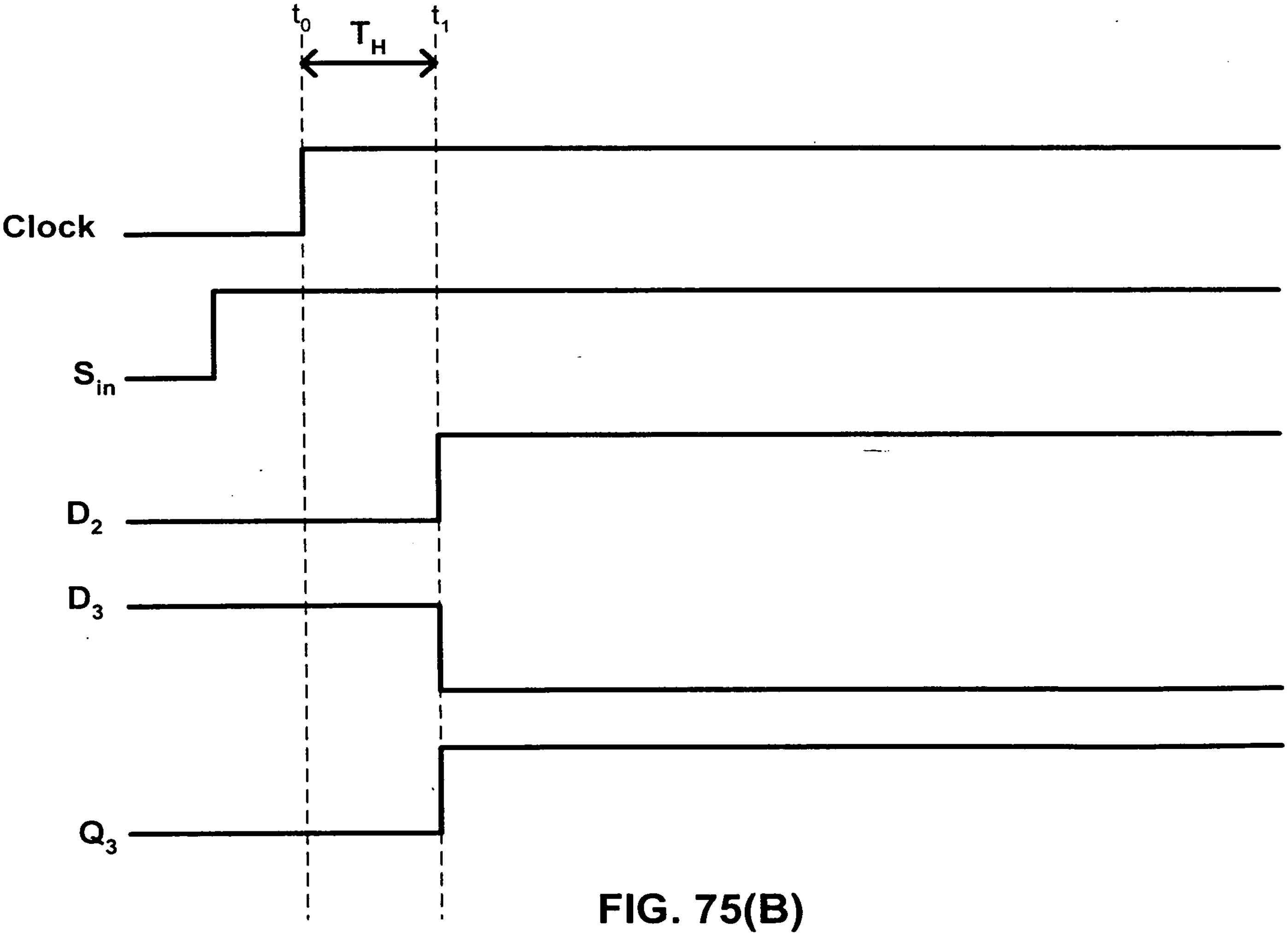


FIG. 75(B)

MULTIPLE FPGA MAPPING
FOR SHIFT REGISTER

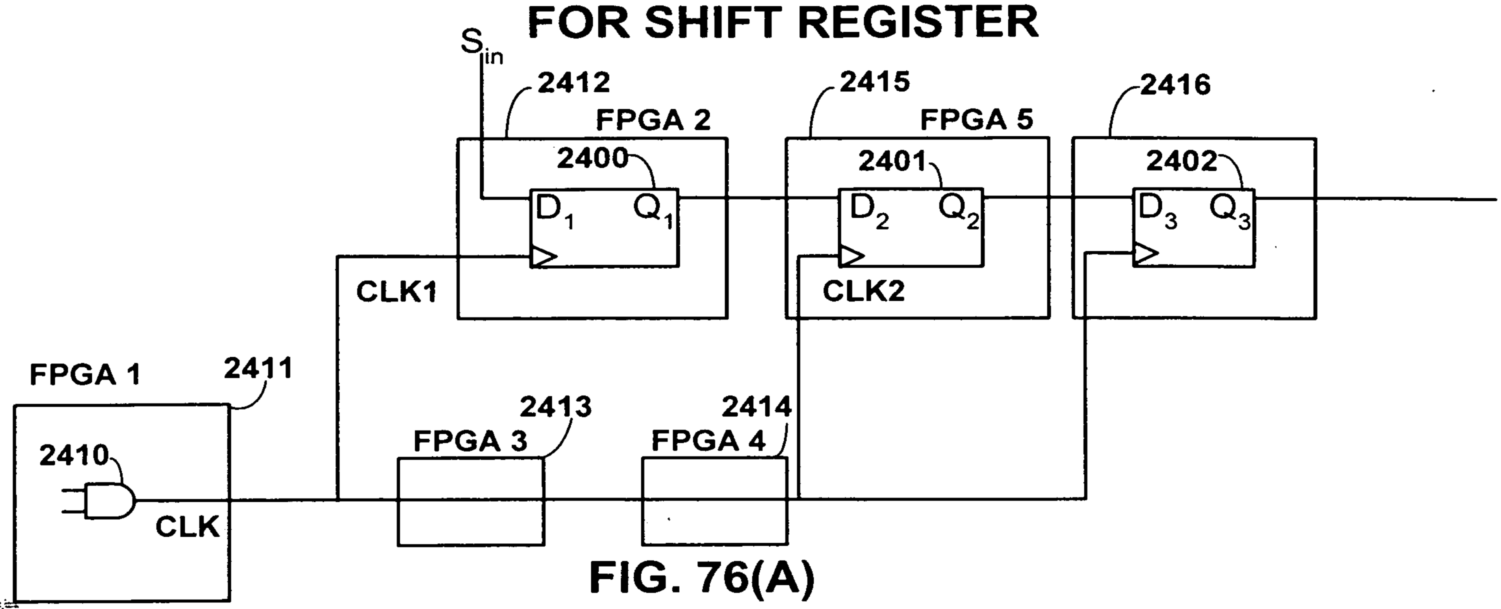


FIG. 76(A)

HOLD TIME VIOLATION
BY LONG CLOCK SKEW

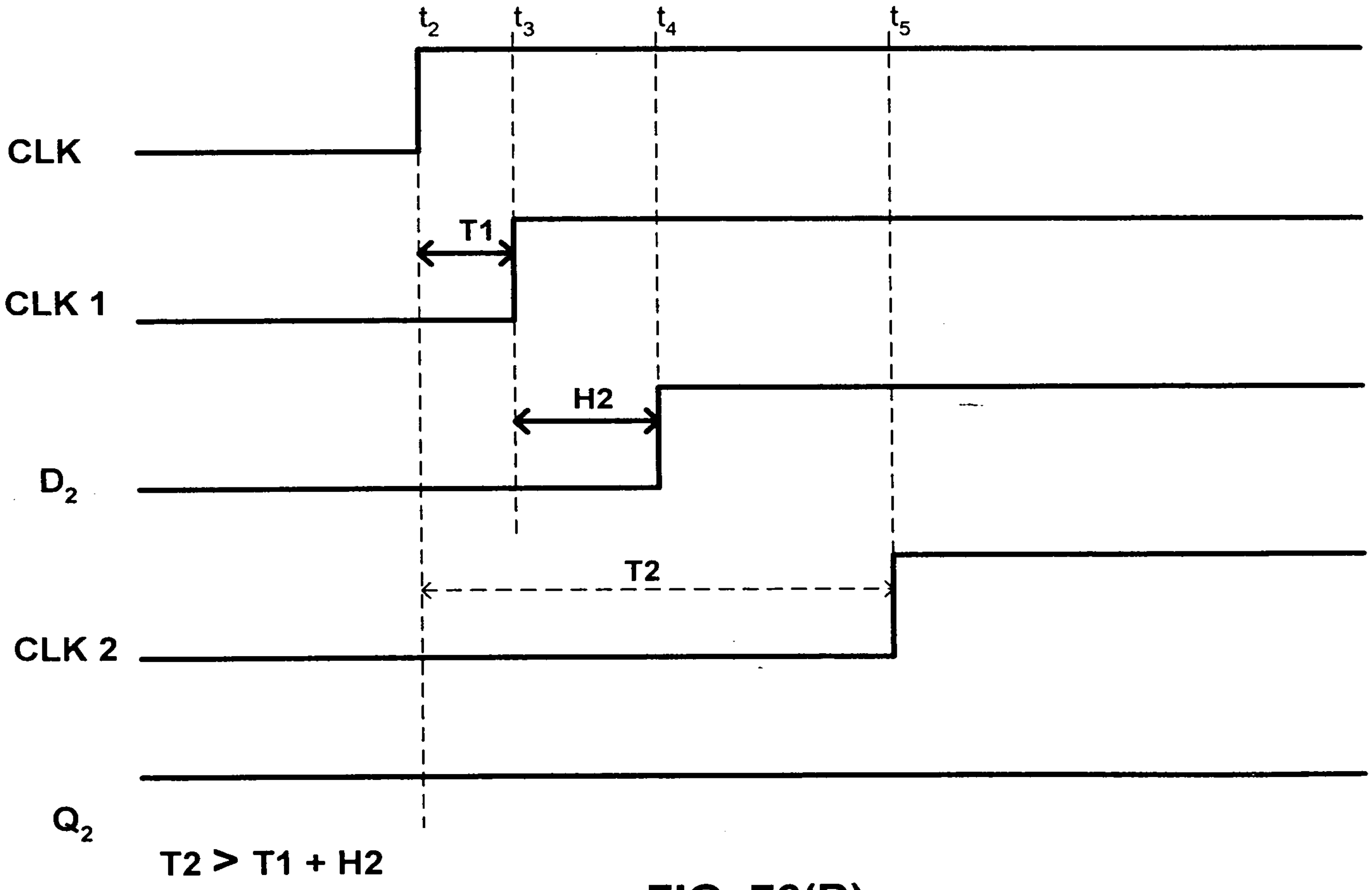


FIG. 76(B)

CLOCK GLITCH PROBLEM

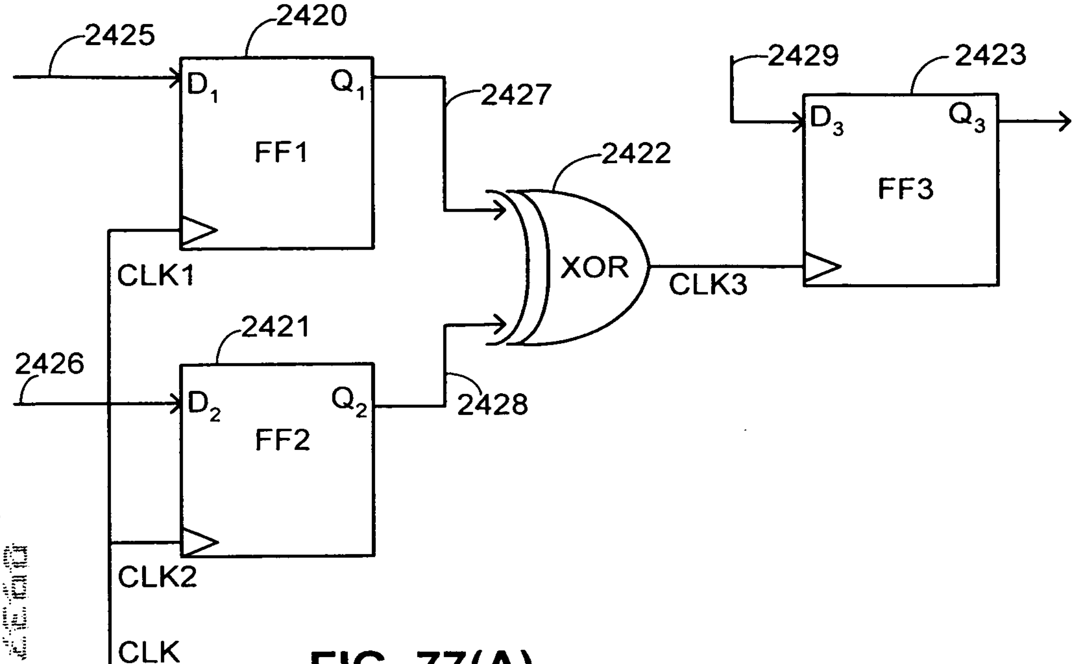


FIG. 77(A)

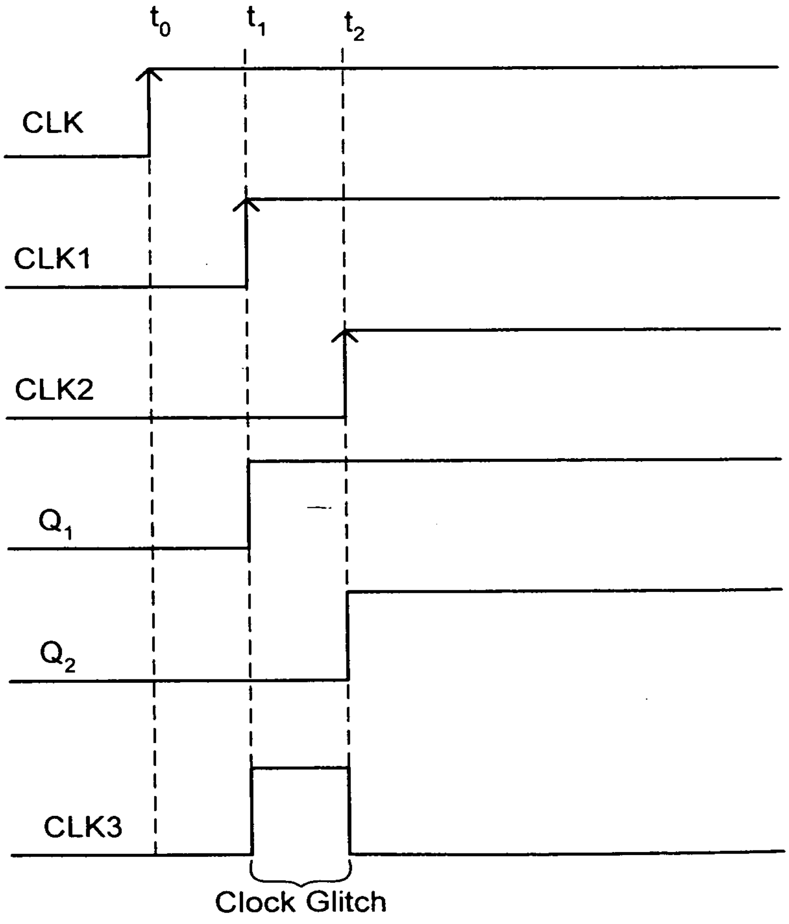
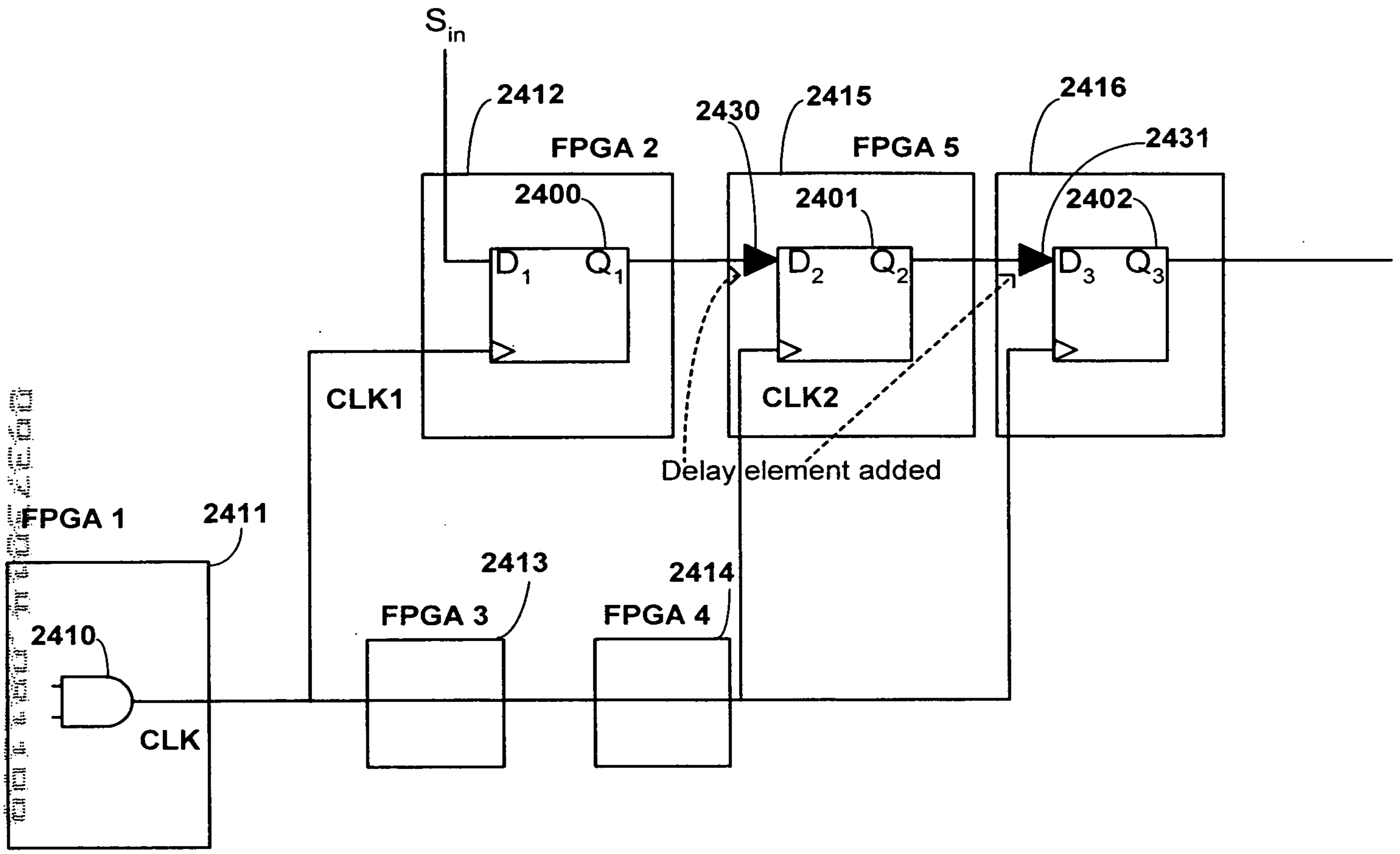


FIG. 77(B)

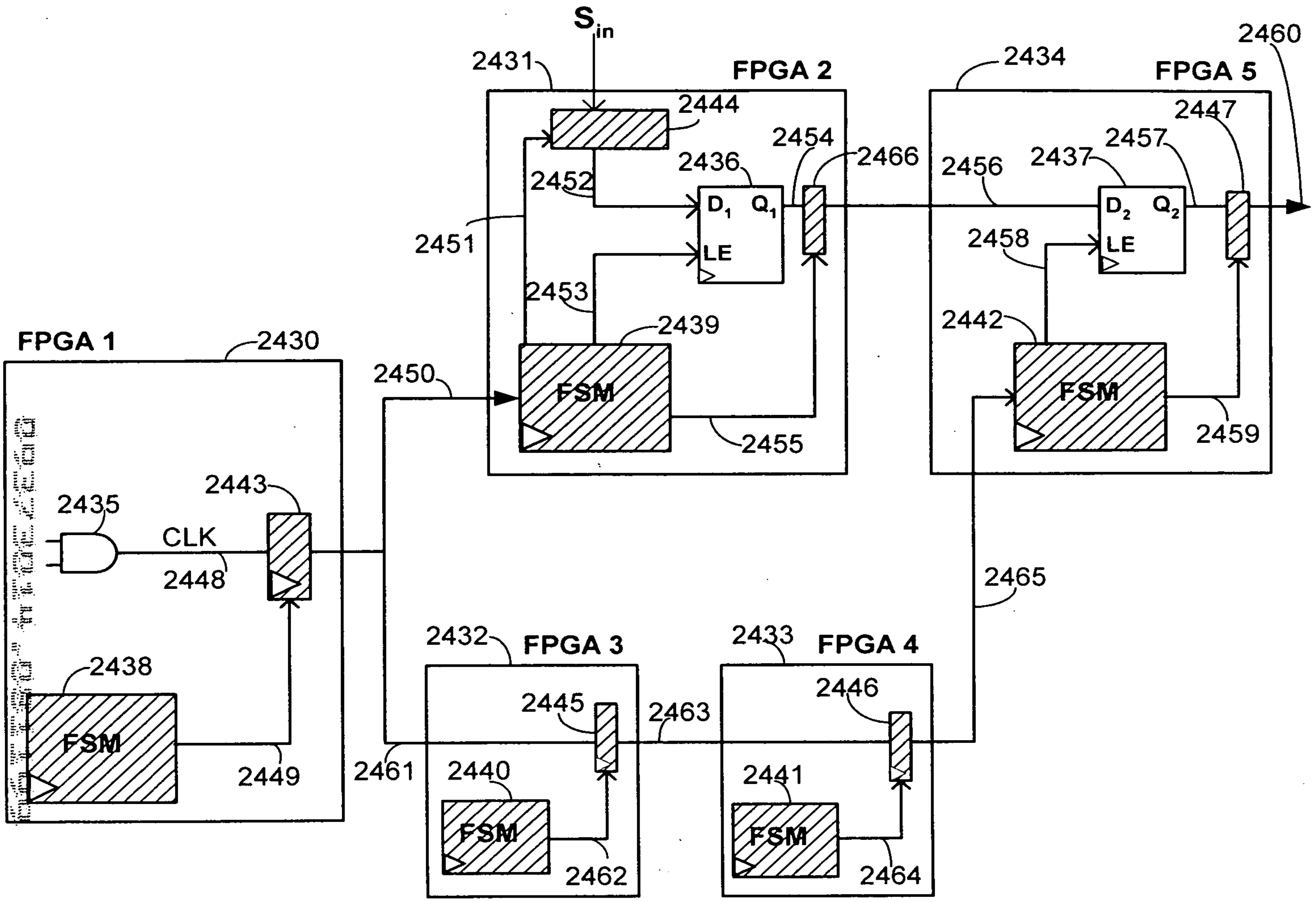
TIMING ADJUSTMENT BY ADDING DELAY



(Prior Art)

FIG. 78

GLOBAL RETIMING



Legend

▷ Controlled by the global reference clock.

▨ FSM and I/O registers for retiming control.

(Prior Art)

FIG. 79

TIGF LATCH

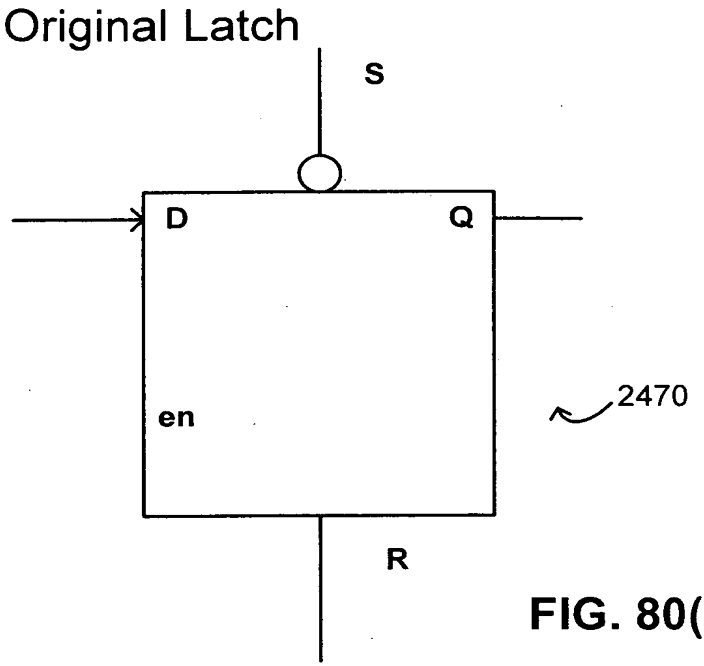


FIG. 80(A)

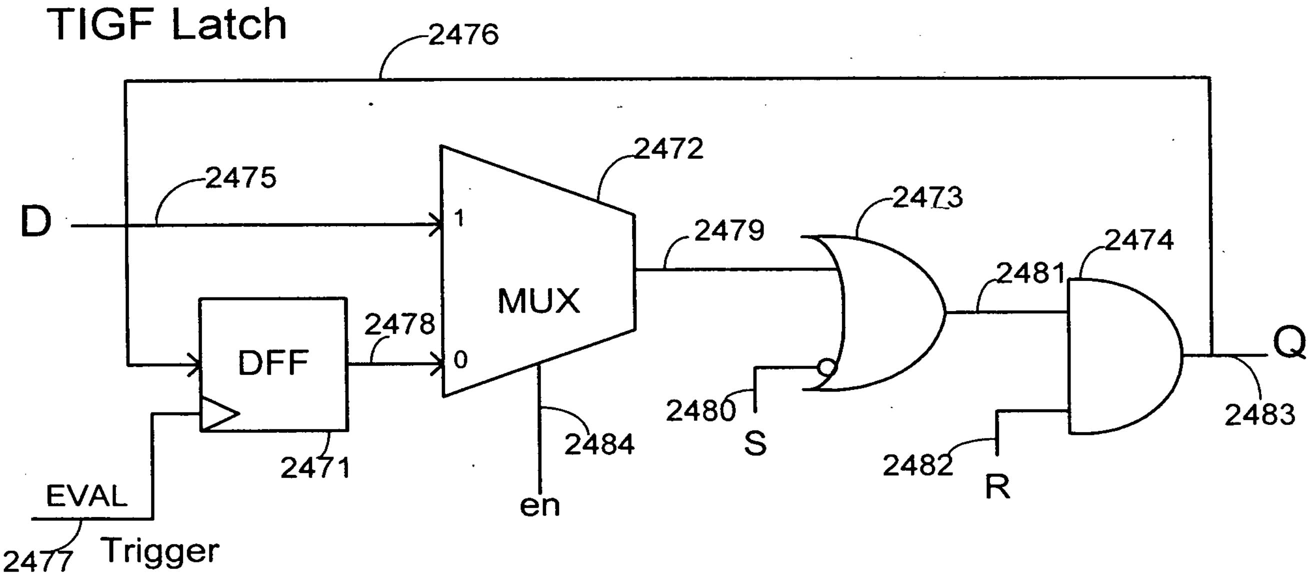


FIG. 80(B)

TIGF DFF

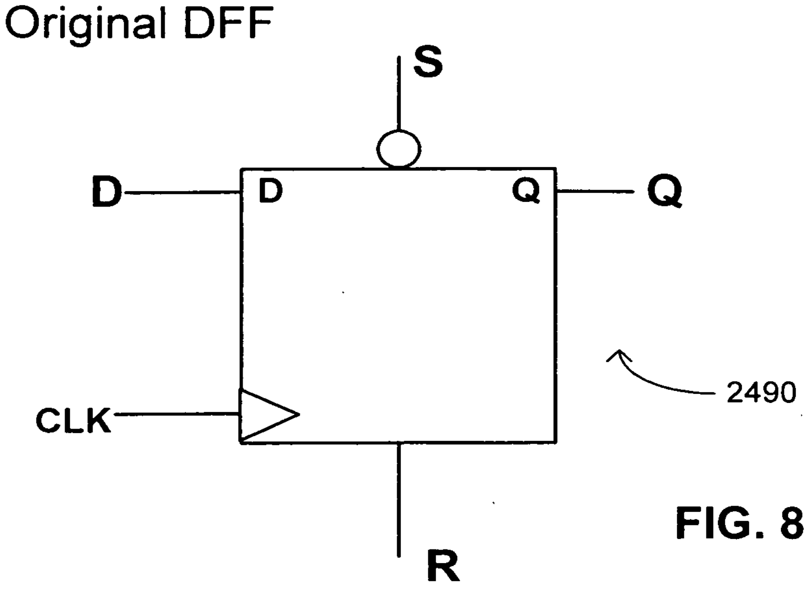


FIG. 81(A)

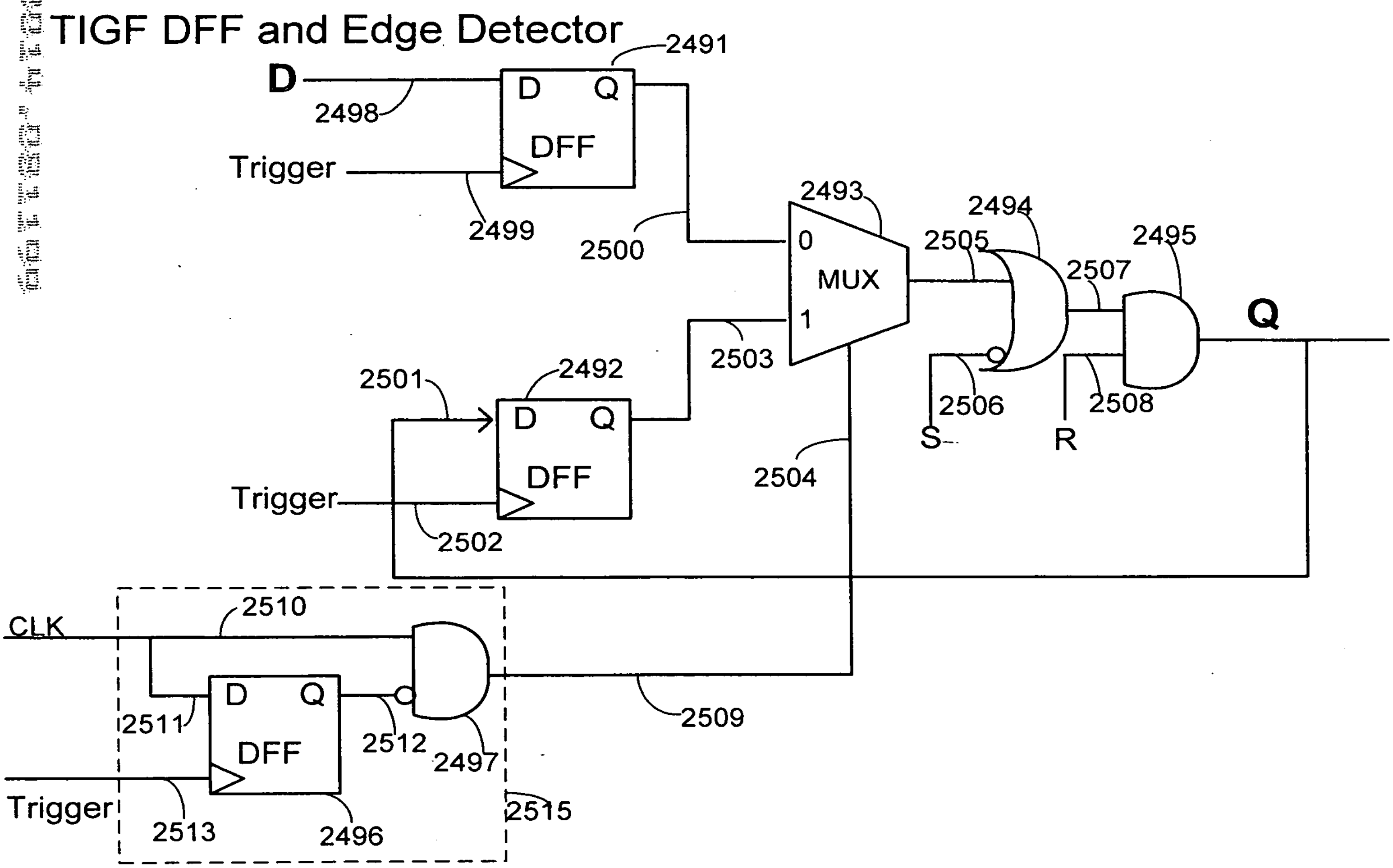


FIG. 81(B)

GLOBAL TRIGGER SIGNAL

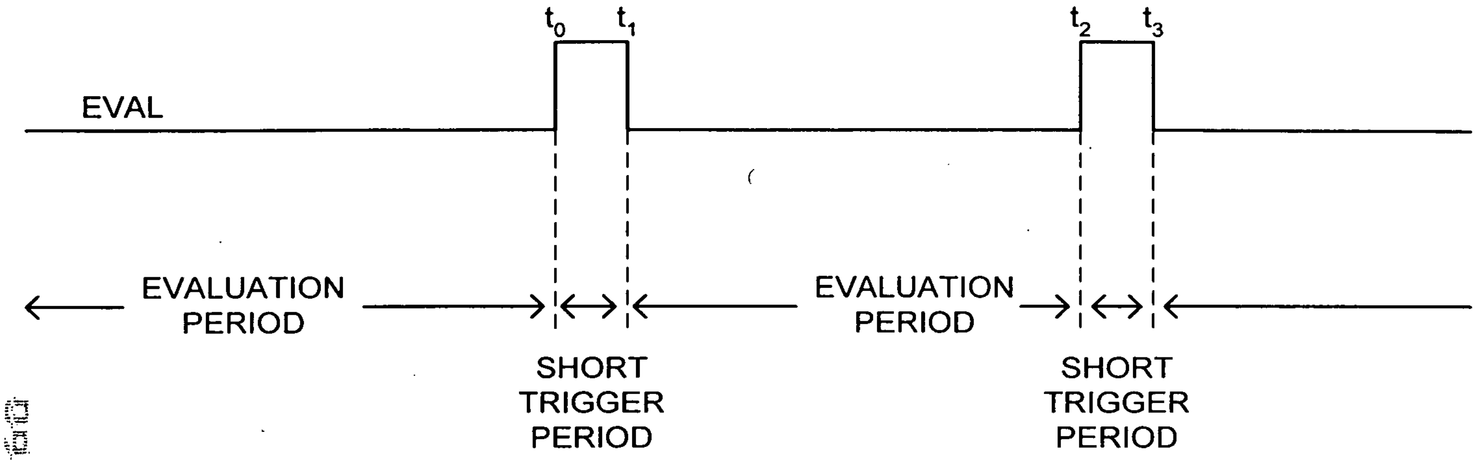


FIG. 82

RCC System

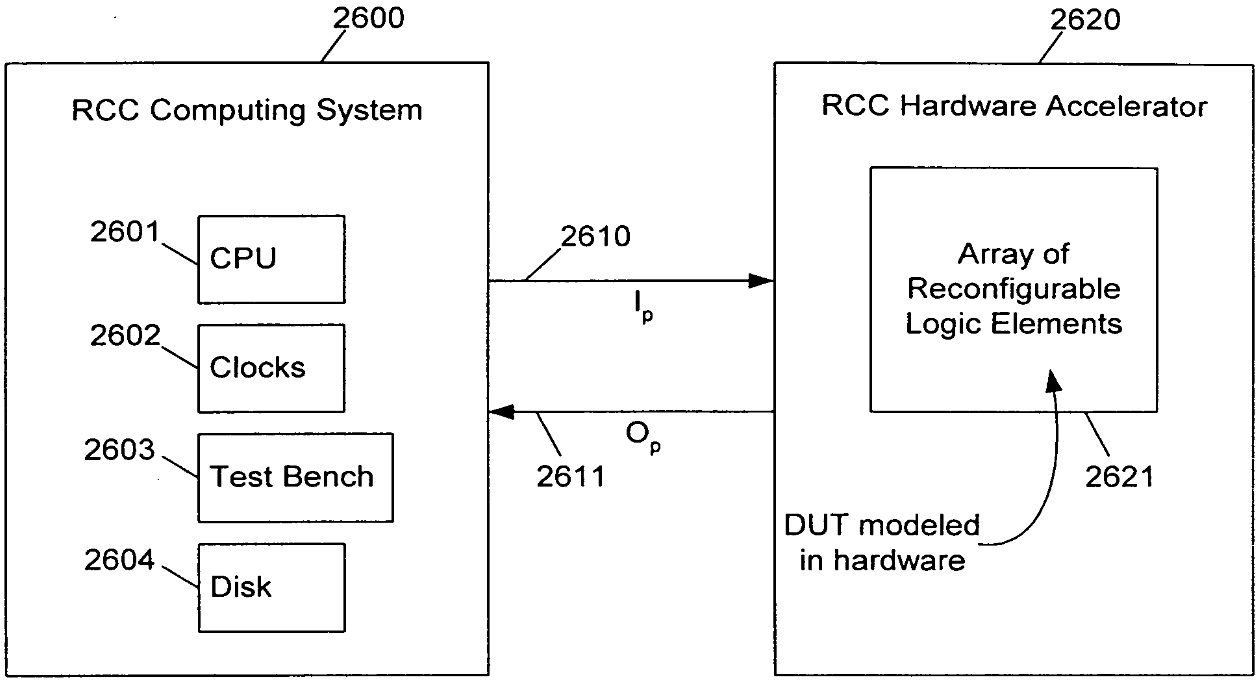


FIG. 83

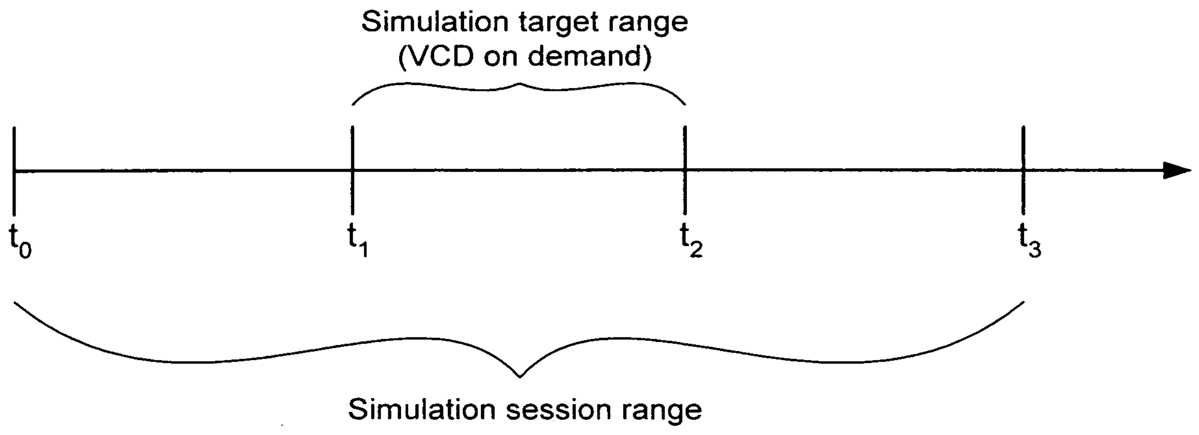

Simulation target range
(VCD on demand)
t_0
t_1
t_2
t_3
Simulation session range

FIG. 84